

ELSA CRISTINA BATISTA BENTO CARVALHO

# PROBABILISTIC CONSTRAINT REASONING

Dissertação apresentada para obtenção do Grau de Doutor em Engenharia Informática, pela Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia.



LISBOA  
2012

---

## **Probabilistic Constraint Reasoning**

Copyright © Elsa Cristina Batista Bento Carvalho, Faculdade de Ciências e Tecnologia,  
Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa tem o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



For *Laura* with love.

*My endless source of inspiration with her tender heart and ingenious mind.*



# Acknowledgements

For me, writing this thesis was like a long, long journey, with many ups and downs. In every step of the way I faced challenges that made me grow as a researcher and as a person. But the journey is coming to an end and is now time to remember and be grateful to those who crossed my path and helped me to achieve my goal in many different ways.

I am deeply indebted to my supervisors, Jorge Cruz and Pedro Barahona. Through their helpful suggestions, they wisely guided me in this process and taught me how to do research. They were always available for mind storming meetings and actively participated in the construction of this work. During the writing of the thesis, their insightful comments were determinant to the final result. Thank you Jorge for your inspired ideas. Thank you Pedro for your careful and knowledgeable revisions.

I am grateful to Dinis Pestana, Sílvia Velosa and Sandra Mendonça for their enlightening comments and discussions in probability theory. Thank you Dinis for revising the chapter of probability theory. Thank you Sílvia for your genuine interest in my work and for your valuable help on some *probabilistic* details. Thank you Sandra for our dinners, where we discussed about probabilities. And thank you for many other things.

I am grateful to Alexandre Goldsztejn for his valuable contribution to this work, while collaborating in the development of verified quadrature methods. The thesis would not be the same without this core component.

I am grateful to Davide D'Alimonte, Guiseppa Zibordi and all his research team in JRC for their collaboration in the ocean color inversion problem. Thank you Davide for your interest in this work and your suggestion of ocean color inversion as an application problem. Without your expert help and guidance it would have been difficult to address this problem. Moreover,

you were fundamental to promote a closer contact with other experts in the area as Guiseppe and his team in JRC. Thank you Guiseppe and Jean-François for disclosing sensitive data, fundamental for our experiments, and for your motivating interest in our proposed solution.

To my colleagues at FCT, Marco, Alexandre and Armando, who are also good friends. They were doing or had already made a similar journey. Their examples made me understand that it was possible to successfully reach the end and their motivating comments were a good aid in difficult moments. To my colleagues at UMa for their patience and valuable support.

To my friends, who were fundamental to remember me that there is life beyond the PhD. Thank you Lulu for your tenderness and joy; Margarida for your solidarity and friendship; Elci, my journey companion, for your sensible advices and solid presence; Cecília for your sense of humor and for caring so much; Professora Rita for being an example to look up to; Elias for your valuable friendship and your inspired mathematical explanations; Leonel for being who you are and Pedro for your enthusiastic support while listening to my research achievements. Thank you Filipe for so many things.

To my parents for their unconditional and loving support, for being there whenever I need them, without criticizing or making questions and for being such a solid home to return to.

To my dear sister that always helped me through the storms. She is a lighthouse in my life.

And to you Laura, my dearest daughter, for waiting so patiently that mum finishes writing her book.

# Abstract

The continuous constraint paradigm has been often used to model safe reasoning in applications where uncertainty arises. Constraint propagation propagates intervals of uncertainty among the variables of the problem, eliminating values that do not belong to any solution. However, constraint programming is very conservative: if initial intervals are wide (reflecting large uncertainty), the obtained safe enclosure of all consistent scenarios may be inadequately wide for decision support. Since all scenarios are considered equally likely, insufficient pruning leads to great inefficiency if some costly decisions may be justified by very unlikely scenarios. Even when probabilistic information is available for the variables of the problem, the continuous constraint paradigm is unable to incorporate and reason with such information. Therefore, it is incapable of distinguishing between different scenarios, based on their likelihoods.

This thesis presents a probabilistic continuous constraint paradigm that associates a probabilistic space to the variables of the problem, enabling probabilistic reasoning to complement the underlying constraint reasoning. Such reasoning is used to address probabilistic queries and requires the computation of multi-dimensional integrals on possibly non linear integration regions. Suitable algorithms for such queries are developed, using safe or approximate integration techniques and relying on methods from continuous constraint programming in order to compute safe covers of the integration region.

The thesis illustrates the adequacy of the probabilistic continuous constraint framework for decision support in nonlinear continuous problems with uncertain information, namely on inverse and reliability problems, two different types of engineering problems where the developed framework is particularly adequate to support decision makers.



# Resumo

O paradigma por restrições em domínios contínuos tem sido amplamente utilizado para modelar raciocínio seguro, em aplicações onde existe incerteza. A propagação de restrições propaga intervalos de incerteza entre as variáveis do problema, eliminando valores que não pertencem a nenhuma solução. No entanto, se os intervalos iniciais forem grandes, a cobertura obtida pode ser demasiado abrangente, sendo insuficiente para suportar decisões. Considerando todos os cenários igualmente verosímeis, tal facto pode resultar em tomadas de decisão de custo elevado devido à existência de cenários muito improváveis. Mesmo quando estão disponíveis distribuições de probabilidade para as variáveis do problema, o paradigma por restrições em domínios contínuos não consegue incorporar nem raciocinar com essa informação.

Neste trabalho desenvolvemos um paradigma de restrições em domínios contínuos que associa um espaço probabilístico às variáveis do problema, permitindo efectuar raciocínio probabilístico. Tal raciocínio baseia-se na avaliação de informação probabilística que requer a computação de integrais multidimensionais em regiões possivelmente não lineares. São desenvolvidos algoritmos capazes de avaliar essa informação, usando técnicas de integração seguras ou aproximadas e dependendo de métodos de programação por restrições em domínios contínuos para obter coberturas da região de integração.

A plataforma probabilística de restrições em domínios contínuos é adequada para suporte à decisão em problemas não lineares em domínios contínuos, com incerteza. A sua aplicabilidade é ilustrada em problemas inversos e problemas de fiabilidade, que são duas classes distintas de problemas de engenharia, representativas do tipo de raciocínio com incerteza requerido pelos decisores.



# Contents

<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Contributions . . . . .	2
1.1.1 Probabilistic Continuous Constraint Space . . . . .	3
1.1.2 Probabilistic Constraint Reasoning . . . . .	3
1.1.3 Prototype Implementation . . . . .	4
1.1.4 Application to Inverse Problems . . . . .	4
1.1.5 Application to Reliability Problems . . . . .	5
1.2 Dissertation Guide . . . . .	5
<b>I Continuous Constraints and Uncertainty</b>	<b>9</b>
<b>2 Continuous Constraint Programming</b>	<b>11</b>
2.1 Interval Analysis . . . . .	11
2.1.1 Interval Arithmetic . . . . .	12
2.1.2 Inclusion Functions . . . . .	18
2.1.3 Interval Methods . . . . .	24
2.2 Continuous Constraint Satisfaction Problems . . . . .	26

## CONTENTS

---

2.3	Computing Feasible Space Approximations . . . . .	30
2.3.1	Constraint Propagation . . . . .	31
2.3.2	Consistencies . . . . .	32
2.3.3	Constraint Reasoning . . . . .	35
2.4	Summary . . . . .	43
<b>3</b>	<b>Probabilistic Uncertainty Quantification</b>	<b>45</b>
3.1	Probability . . . . .	45
3.2	Conditional Probability . . . . .	49
3.3	Random Variables . . . . .	53
3.3.1	Moments . . . . .	56
3.3.2	Some Continuous Probability Distributions . . . . .	58
3.4	Random Vectors . . . . .	59
3.4.1	Moments . . . . .	62
3.4.2	Conditioning . . . . .	63
3.5	Numerical Computations . . . . .	65
3.5.1	Probabilistic Framework Outline . . . . .	66
3.5.2	Integration with Taylor Models . . . . .	66
3.5.3	Integration with Monte Carlo . . . . .	72
3.6	Summary . . . . .	75
<b>II</b>	<b>Probabilistic Constraints</b>	<b>77</b>
<b>4</b>	<b>Probabilistic Constraint Programming</b>	<b>79</b>
4.1	Probabilistic Continuous Constraint Space . . . . .	81
4.2	Probabilistic Constraint Events . . . . .	83
4.3	Safe Integral Enclosure . . . . .	88

4.4	Probability Enclosure . . . . .	90
4.5	Conditional Probability Enclosure . . . . .	93
4.6	Algorithms . . . . .	96
4.6.1	Probability Enclosure . . . . .	97
4.6.2	Conditional Probability Enclosure . . . . .	101
4.7	Alternative Approximate Computations . . . . .	105
4.8	Experimental Results . . . . .	107
4.8.1	Development Environment . . . . .	107
4.8.2	The PC Events . . . . .	108
4.8.3	Probability Enclosure . . . . .	110
4.8.4	Conditional Probability Enclosure . . . . .	112
4.9	Summary . . . . .	113
<b>5</b>	<b>Random Vectors</b>	<b>115</b>
5.1	Probabilistic Enclosures . . . . .	115
5.2	Conditional Probabilistic Enclosures . . . . .	124
5.3	Algorithms . . . . .	129
5.4	Probability Distributions . . . . .	131
5.4.1	Probability Distributions for Functions of Random Vectors . . . . .	135
5.5	Alternative Approximate Computations . . . . .	140
5.6	Experimental Results . . . . .	141
5.6.1	The PC Events . . . . .	141
5.6.2	Probabilistic Enclosures . . . . .	142
5.6.3	Probability Distributions . . . . .	144
5.6.4	Conditional Probabilistic Enclosures . . . . .	150
5.7	Summary . . . . .	151

## CONTENTS

---

<b>III</b>	<b>Application to Decision Problems</b>	<b>153</b>
<b>6</b>	<b>Nonlinear Inverse Problems</b>	<b>155</b>
6.1	Inverse Problems . . . . .	156
6.2	Classical Techniques . . . . .	157
6.3	Probabilistic Constraint Approach . . . . .	163
6.4	Seismic Event Model . . . . .	168
6.5	Population Growth Model . . . . .	169
6.6	Ocean Color Inversion . . . . .	175
6.6.1	Ocean Color . . . . .	176
6.6.2	Related Work . . . . .	177
6.6.3	Probabilistic Constraint Approach . . . . .	178
6.6.4	Experimental Results . . . . .	180
6.7	Summary . . . . .	186
<b>7</b>	<b>Reliability Problems</b>	<b>187</b>
7.1	Reliability Analysis . . . . .	189
7.1.1	Reliability Based Design . . . . .	190
7.2	Classical Techniques . . . . .	192
7.3	Probabilistic Constraint Approach . . . . .	197
7.3.1	Reliability Assessment . . . . .	197
7.3.2	Reliability Based Design . . . . .	204
7.3.3	Reliability Based Design Optimization . . . . .	213
7.4	Short Rectangular Column . . . . .	219
7.5	Summary . . . . .	223

---

<b>8</b>	<b>Conclusions and Future Work</b>	<b>225</b>
8.1	Probabilistic Constraint Programming . . . . .	225
8.2	Prototype . . . . .	228
8.3	Application to Decision Problems . . . . .	229
	<b>Appendices</b>	<b>233</b>
<b>A</b>	<b>Constraint Reasoning Algorithm</b>	<b>233</b>
A.1	Parametrization . . . . .	233
A.2	Algorithm Convergence . . . . .	237
<b>B</b>	<b>Integration with Taylor Models</b>	<b>241</b>
B.1	Proof of Property 3.16 . . . . .	241
B.2	Numerical Computations of Example 3.2 . . . . .	244
	<b>References</b>	<b>247</b>

## CONTENTS

---

# List of Figures

2.1	Model and propagate uncertainty. (a) Model uncertainty in the measurement of the radius. (b) Propagation of uncertainty to the calculus of the diameter. . . . .	12
2.2	Enclosure of the area exact value: $3 \leq area = \pi \leq 4$ . . . . .	14
2.3	Joint box covers of the feasible space inside the <i>curve lines</i> . . . . .	38
2.4	Joint box covers obtained when computing the volume of a circle with radius 1, centered in (0,0). . . . .	43
3.1	(a) Bivariate Gaussian distributed probability space; (b) Conditional probability space distribution given the evidence $x - 1 \leq y \leq x + 1$ . . .	52
3.2	Mapping from the sample space $\Omega$ to $\mathbb{R}$ , by the random variable $X$ . . . . .	53
4.1	(a) Joint PDF of $\mathbf{X}$ (b) Constraints. . . . .	83
4.2	Some possible events of a probability space defined by a PC. . . . .	85
4.3	Boxes and constraint . . . . .	86
4.4	Events and PDF used in examples. . . . .	98
4.5	Joint box covers of the PC event $\mathcal{H}_1$ computed when using algorithm 6 with a safe integration method. The boundary boxes are light gray and the inner boxes are gray. . . . .	99
4.6	Joint box covers computed when using algorithm 7 with $\delta = 0.05$ . The common/other boundary boxes are light blue/gray, the common/other inner boxes are dark blue/gray. . . . .	105

## LIST OF FIGURES

---

4.7	Joint box covers of the PC event $\mathcal{H}_1$ computed when using algorithm 6 with an approximate integration method. The boundary boxes are light gray and the inner boxes are gray. . . . .	106
4.8	Probability density functions used for testing. . . . .	108
4.9	Events used for testing. . . . .	109
5.1	PDF and box cover used in examples. . . . .	117
5.2	Restricting the values of random variables. . . . .	119
5.3	Integrand functions. . . . .	121
5.4	$\mathcal{H}_5$ and its joint box cover $\mathcal{H}_{\boxplus}$ . . . . .	125
5.5	Event $\mathcal{H}$ and joint box covers of $D$ and $\mathcal{H}$ computed with algorithms 10 and 11 with a safe integration method. The boundary boxes are light gray and the inner boxes are gray. . . . .	131
5.6	Integrand functions used to computed the expected values and variances. . . . .	132
5.7	Joint box covers produced when computing probability distributions. . . . .	136
5.8	Marginal probability distributions. . . . .	136
5.9	Joint box cover of $\mathcal{H}$ (top) and correspondent joint box cover of $\mathbf{Y}$ (bottom). . . . .	137
5.10	Joint box cover and distribution produced by algorithm 13. . . . .	140
5.11	Probability density functions used for testing. . . . .	141
5.12	Events used for testing. . . . .	142
5.13	Probability distributions of $\mathbf{Z} = \langle X, Y \rangle$ conditioned by events $\mathcal{H}_i$ obtained with algorithm 12 with PDF $f_1$ . . . . .	145
5.14	Probability distributions of $\mathbf{Z} = \langle X, Y \rangle$ conditioned by events $\mathcal{H}_i$ obtained with algorithm 12 with PDF $f_2$ . . . . .	145
5.15	Marginal probability distributions of $X$ and $Y$ conditioned by events $\mathcal{H}_i$ obtained with algorithm 12 with PDF $f_1$ . . . . .	146
5.16	Marginal probability distributions of $X$ and $Y$ conditioned by events $\mathcal{H}_i$ obtained with algorithm 12 with PDF $f_2$ . . . . .	146

5.17	Probability distributions of $XY$ conditioned by PC events obtained with algorithm 12 with PDF $f_1$ . . . . .	147
5.18	Probability distributions of $XY$ conditioned by PC events obtained with algorithm 12 with PDF $f_2$ . . . . .	148
5.19	Probability distributions conditioned by $D$ obtained with algorithm 12 with PDF $f_1$ . . . . .	148
5.20	Probability distributions conditioned by $D$ obtained with algorithm 12 with PDF $f_2$ . . . . .	149
5.21	Probability distribution of $\langle X + Y, X \rangle   D$ with PDF $f_2$ . . . . .	149
6.1	Least squares approaches a) search for a single best-fit solution that maximize the likelihood of the observations. Bounded-error approaches b) compute a set of solutions consistent with the observations. . . . .	159
6.2	In linear problems, least squares approaches a) compute the mean value of the distribution of the parameter values given the observed data. Bounded-error approaches b) compute an interval that includes all $m$ -values consistent with the observations. . . . .	160
6.3	In nonlinear problems, least squares approaches a) may provide wrong results assuming the computed value to be the mean of a parameter normal distribution given the observed data. Bounded-error approaches b) still provide reliable enclosures for all $m$ -values consistent with the observations. . . . .	162
6.4	Probability distribution of the parameter $m$ obtained by the PC framework.	167
6.5	Epicentral coordinates of the seismic event. Joint distribution. . . . .	170
6.6	Epicentral coordinates of the seismic event. Marginal distributions (a) for $m_1$ and (b) for $m_2$ . . . . .	171
6.7	Enclosure of the CCSP feasible space. . . . .	172
6.8	Exponential model. (a) Joint distribution; Marginal distributions (b) for $m_1$ and (c) for $m_2$ . . . . .	173

## LIST OF FIGURES

---

6.9	Logistic model. Marginal distributions (a) for $m_1$ , (b) for $m_2$ and (c) for $m_3$ . . . . .	174
6.10	Expected US population in 1920. (a) Exponential and (b) logistic models.	174
6.11	The forward model is a function from the optically active seawater compounds ( $Chla$ , $NPPM$ and $CDOM$ ) to the remote sensing reflectance ( $R_{rs}$ ) at a given wavelength ( $\lambda$ ). . . . .	177
6.12	Joint and marginal uncertainty distributions computed by the PC framework. . . . .	182
7.1	$\mathbf{X}$ space and $\mathbf{U}$ space. . . . .	193
7.2	Geometrical illustration of the reliability index. . . . .	193
7.3	Problems with multiple design points. . . . .	195
7.4	Example of a linear limit-state in the original space and non-linear in the standard normal space. . . . .	199
7.5	Non linear limit-state function $g(x_1, x_2) = 0$ . . . . .	201
7.6	Examples of series and parallel systems found in [111]. . . . .	202
7.7	A design problem. . . . .	206
7.8	Reliability distribution over the design space. . . . .	210
7.9	Reliability distribution over the design space. . . . .	213
7.10	Reliability based design optimization. . . . .	216
7.11	Reliability based design optimization. . . . .	217
7.12	Multiple optimal designs. . . . .	218
7.13	Pareto-optimal frontier . . . . .	218
7.14	Reliability distribution over the design space for problem ColumnB. . .	221
7.15	Plots of $g(2000, 500, 5, h, b)$ (red line) and $g(2000, 500, e^5, h, b)$ (green line).	222
7.16	Reliability distribution over the design space for problem ColumnC. . .	222
7.17	Reliability distribution over the design space for problem ColumnD. . .	223

# List of Tables

3.1	Enclosure for the quadrature of $f$ in $B$ by Taylor models integration with increasing order $n$ . . . . .	72
3.2	Estimated integral of $f$ in $B$ by Monte Carlo integration and correspondent error estimate. . . . .	75
4.1	Probabilities of $B_i$ and $B_i \cap \mathcal{H}_5$ using formulas (4.1) and (4.2). . . . .	87
4.2	Enclosures for the probability of $B_i \cap \mathcal{H}_5$ . . . . .	91
4.3	Definition of constraints and events. . . . .	98
4.4	Definition of constraints and events. . . . .	108
4.5	Probability enclosures for events $\mathcal{H}_i$ with PDF $f_1$ , obtained with both versions of algorithm 6, their numerical computations with <i>Mathematica</i> and respective timings. . . . .	111
4.6	Probability enclosures for events $\mathcal{H}_i$ with PDF $f_2$ , obtained with both versions of algorithm 6, their numerical computations with <i>Mathematica</i> and respective timings. . . . .	111
4.7	Midpoints of the conditional probability enclosures for $\mathcal{H}$ given $\mathcal{H}_i$ obtained by the PCTM version of algorithm 7 and respective timings (in seconds). . . . .	112
4.8	Relative error percentages of the conditional probabilities of $\mathcal{H}$ given $\mathcal{H}_i$ obtained with <i>Mathematica</i> when compare with those computed by algorithm 7. . . . .	112

## LIST OF TABLES

---

4.9	Speedup of algorithm 7 when compared with the alternative method (with algorithm 6) that uses two completely independent joint box covers.	113
5.1	Probability enclosures of $B_i$ using property 4.4. . . . .	118
5.2	Enclosures for the probability of $B_i \cap \mathcal{H}_j$ . . . . .	118
5.3	Enclosures for the integrals $\int_B x_j f(\mathbf{x}) d\mathbf{x}$ , $\int_B x_j^2 f(\mathbf{x}) d\mathbf{x}$ and $\int_B x_1 x_2 f(\mathbf{x}) d\mathbf{x}$ .	122
5.4	Exact values of $\int_B x_j f(\mathbf{x}) d\mathbf{x}$ , $\int_B x_j^2 f(\mathbf{x}) d\mathbf{x}$ and $\int_B x_1 x_2 f(\mathbf{x}) d\mathbf{x}$ . . . . .	123
5.5	Enclosures for the probability of $B_i \cap \mathcal{H}_j \cap \mathcal{H}_5$ . . . . .	125
5.6	Exact values for the integrals over the region defined by $\mathcal{H}_5$ . . . . .	128
5.7	Definition of constraints and events. . . . .	143
5.8	Probability enclosures for events $\mathcal{H}_i$ with PDF $f_1$ , obtained with both versions of algorithm 6 and respective timings. . . . .	143
5.9	Probability enclosures for events $\mathcal{H}_i$ with PDF $f_2$ , obtained with both versions of algorithm 6 and respective timings. . . . .	143
5.10	Expected values and variances conditioned by events $\mathcal{H}_i$ obtained by PCTM and PCMC versions, with PDF $f_1$ . . . . .	150
5.11	Enclosures for expected values and covariances conditioned by events $\mathcal{H}_i$ obtained with PCTM version, with PDF $f_2$ . . . . .	150
5.12	Approximations for expected values and covariances conditioned by events $\mathcal{H}_i$ obtained with PCMC version, with PDF $f_2$ . . . . .	151
6.1	Arrival times (in seconds) of the seismic waves observed at six seismic stations. . . . .	168
6.2	US Population (in millions) over the years 1790 (0) to 1910 (120). . . . .	170
6.3	Matrix of coefficients computed from the forward model of figure 6.11 and the values for the constants. . . . .	179
6.4	The 12 experimental cases were simulated by using the forward model to compute the observed parameter values from the above model parameters values. . . . .	180

## LIST OF TABLES

---

6.5	Interval enclosures of $E[Chla]$ and $STD[Chla]$ computed by the PCTM version of algorithm 11. . . . .	182
6.6	Approximations of $E[Chla]$ and $STD[Chla]$ computed by the PCMC version of algorithm 11. . . . .	183
6.7	Approximations of $E[Chla]$ and $STD[Chla]$ computed by the AMC algorithm after 10 minutes with several sampling sizes $N$ . . . . .	184
6.8	$E[Chla]$ and $STD[Chla]$ obtained by the PC algorithms for different accuracies. . . . .	186
7.1	Probability of failure $\times 10^2$ . . . . .	200
7.2	Probability of failure $\times 10^2$ . . . . .	201
7.3	Probability of failure $\times 10^3$ . . . . .	203
7.4	Probability of failure $\times 10^2$ . . . . .	204
7.5	Reliability values for some particular designs. . . . .	211

## LIST OF TABLES

---

## CHAPTER 1

# Introduction

A mathematical model typically describes a system by a set of variables and a set of constraints that establish relations between the variables. In this thesis we focus on continuous domains, i.e. variables ranging over intervals of real numbers and relations defined on these intervals. Uncertainty and non linearity play a major role in modeling most real world continuous systems. When the model is non linear small approximation errors may be dramatically magnified. Any framework for decision support in continuous domains must provide an expressive mathematical model to represent the system behavior performing sound reasoning that accounts for the uncertainty and the effect of non linearity. Two classical approaches to reason with uncertainty exploit different scenarios consistent with the mathematical model.

When safety is a major concern, all possible scenarios must be considered. For this purpose, intervals can be used to include all possible values of the variables. This is the approach adopted in continuous constraint programming which uses safe constraint propagation techniques to narrow the intervals, thus reducing uncertainty. Nevertheless this approach considers all the scenarios to be equally likely, leading to great inefficiency: some costly decisions may be taken due to very unlikely scenarios.

In contrast, stochastic approaches reason on approximations of the most likely scenarios. They associate a probabilistic model to the problem thus characterizing the likelihood of the different scenarios. Some methods use local search techniques to find most likely scenarios, which may lead to erroneous decisions due to approximation errors, non linearity as well as the inherent incompleteness of such type of search. Moreover, there may be other scenarios relevant to decision making which are ignored by this single

scenario approach. Other stochastic methods use extensive random sampling over the different scenarios to characterize the complete probability space. However, even after intensive computations, no safe conclusion can be drawn from these methods, because a significant subset of the probabilistic space may have been missed.

The fundamental motivation of this work is thus to extend the continuous constraint framework with enough expressive power so as to allow the representation of uncertain information characterized by probability distributions. This will clearly broaden the range of problems that can be modeled in the continuous constraint paradigm, making it a more appealing tool for decision support in engineering and science, where problems in continuous domains with uncertainty and non linearity are common.

This work extends the classical continuous constraint approach, by complementing the interval representation of uncertainty with a probabilistic characterization of the distribution of possible values. A new formalism is thus available, allowing to reason with probabilistic information, while maintaining all the advantages (e.g. safety) of continuous constraint reasoning, producing probability enhanced intervals for the acceptable values of the problem variables. Since probabilistic reasoning in continuous domains involves multi dimensional integral computations, both safe and approximate techniques to obtain such integral values were addressed.

To assess the adequacy of the proposed formalism, the probabilistic constraint framework was applied to several decision support problems in continuous domains, in the presence of stochastic uncertainty and non linearity by means of a prototype that implemented all the algorithms presented in the thesis.

Two main classes of problems that can benefit with the proposed techniques, where uncertainty and non linearity are present were exploited: inverse problems and reliability problems. A comprehensive comparison of the probabilistic constraint framework with classical techniques that address those problems was performed.

## 1.1 Contributions

The main contributions of this thesis can be summarized as follows.

- Extension of the continuous constraint paradigm to handle and reason with probabilistic information, providing a new formalism to model continuous constraint problems that includes probability distributions for the variables of the problem.
- Theoretically characterization of the framework that performs probabilistic constraint reasoning and address its operational aspects.
- Development of a prototype, that implements the operational aspects of the probabilistic constraint framework, to test its capabilities as an alternative approach to decision support in the presence of stochastic uncertainty and non linearity.
- Illustration of the adequacy and potential of the framework by applying the prototype to inverse problems and reliability problems.

### 1.1.1 Probabilistic Continuous Constraint Space

A new formalism, the probabilistic continuous constraint space, PC, was defined that incorporates information on probability distributions of the variables of the problem, a joint probability density function. This associates a probabilistic space to a continuous constraint satisfaction problem, CCSP, allowing to formulate several probabilistic queries and obtain a set of probabilistic data associated with the problem.

The two basic components of probability theory, events and random variables, are identified within a probabilistic continuous constraint space. PC events are represented by regions that satisfy (subsets of) the constraints of the underlying CCSP. The random variables are represented by the CCSP variables which now have an associated joint probability density function.

### 1.1.2 Probabilistic Constraint Reasoning

Probabilistic constraint reasoning, relies on constraint reasoning to cover the events with adequate sets of boxes, and on integration techniques to compute probabilities of events, expected values and variances of random variables.

The representation of events by sets of boxes is twofold: it aims at transforming a complex problem (integration on a non linear region) into simpler problems (integration on boxes) and to provide guaranteed bounds (when safe integration techniques

are available) for the computed quantities by distinguishing between boxes completely included in the event and those that are not.

In this context, a fundamental operational issue in probabilistic constraint reasoning is the computation of multi dimensional integrals in boxes. In contrast to approximate numerical methods, a more powerful technique was developed (with Alexandre Goldsztejn) to obtain safe integration bounds. On the other hand, trading safety with efficiency, an approximate Monte Carlo integration hybrid technique was implemented, benefiting from the pruning of the sampling space with constraint programming.

### 1.1.3 Prototype Implementation

Besides the theoretical formalization of the proposed probabilistic constraint framework, its operational behavior is exploited and tested in a prototype implementation.

Since continuous constraint programming is a key aspect of the framework, this prototype was implemented over RealPaver 1.0, a state-of-the-art interval solver, that provides methods from continuous constraint programming and allows to easily extend them with probabilistic reasoning.

Several methods were thus implemented that, given one or more files describing the problems (e.g. PC events, probabilistic continuous constraint spaces) compute different types of probabilistic results: conditional and unconditional probabilities of PC events, conditional and unconditional expected values and covariance matrices of random vectors and parametric probability distributions of random vectors. A simple optimization algorithm is also provided that allows to compute Pareto-optimal frontiers given multi objective problems.

### 1.1.4 Application to Inverse Problems

Inverse problems aim to estimate parameters of a model of the system behavior from observed data. Uncertainty arises from measurement errors on the observed data or approximations in the model specification. Non linear model equations can cause a severe amplification of the uncertainty, and an arbitrarily small change in the data may induce an arbitrarily large change in the values of the model parameters.

The probabilistic continuous constraint approach overcomes many problems posed by classical techniques. It associates *a priori* probabilistic information to the observed data range of values, produces a characterization of the complete solution space in terms of *a posteriori* probability distributions for the range of values of the model parameters given the evidence provided by the constraints, and guarantees robustness of the solutions found when safe integration techniques are used. This allows decision makers to compare different values for sets of model parameters based on their likelihoods, instead of being artificially restricted to a single solution or to be confronted with several solutions without being able to differentiate between them.

### 1.1.5 Application to Reliability Problems

Reliability problems aim to find reliable decisions according to a model of the system behavior, where both decision and uncontrollable variables may be subject to uncertainty. Given the choices committed in a decision, its reliability quantifies the ability of the system to perform the required functions under variable conditions. Many reliability problems include optimization criteria, modeled by objective functions over both the uncontrollable and decision variables.

The paradigm proposed in this thesis does not suffer from the limitations of classical methods, guaranteeing safe bounds for the reliability of a decision. Furthermore, where classical techniques only provide a single decision point for a given target reliability, the proposed approach provides a global view of the whole decision space reliability. For example, in reliability based optimization, our method computes a safe Pareto-optimal frontier given the optimization criteria and the maximization of the reliability value, providing valuable information on the tradeoff between the system reliability value and its desired behavior.

## 1.2 Dissertation Guide

This dissertation is divided in three main parts.

The first part describes the relevant concepts of continuous constraint programming and probability theory and is composed of chapters 2 and 3.

Chapter 2 focuses on the main concepts of continuous constraint programming, by introducing interval analysis (interval arithmetic, inclusion functions and interval methods), continuous constraint satisfaction problems and methods to solve them (constraint propagation and constraint reasoning). Although describing classical notions and techniques, the topics are biased towards the probabilistic continuous constraint framework proposed in the thesis.

Chapter 3 introduces relevant concepts of probability theory, presenting unconditional and conditional probabilities, random variables and random vectors, in a continuous probability space. In this context, two numerical integration methods are presented to compute either safe or approximate enclosures for exact integral values: Taylor Models and Monte Carlo methods, respectively.

The second part addresses probabilistic constraint reasoning, defining a formalism to represent probabilistic information in continuous constraint problems and describing techniques to reason with this new information and compute probabilistic data. It is composed of chapters 4 and 5.

Chapter 4 defines the probabilistic continuous constraint space and its semantics, and identifies the kind of problems that can be formulated within this probabilistic space. The concept of probabilistic constraint event is introduced and both safe and approximate methods to obtain enclosures for their unconditional and conditional probabilities are described. Experimental results illustrate the capabilities of the proposed algorithms.

Chapter 5 addresses the probabilistic features of random vectors within a probabilistic continuous constraint space. It presents methods to compute safe and approximate enclosures for probabilities (conditional or not) of random vectors when restricted to a range of values, as well as for unconditional and conditional expected values and covariance matrices of random vectors. It also presents methods to compute probability distributions of a subset of the identity random vector and of random vectors defined as functions of the former. Experimental results illustrate the capabilities of the proposed algorithms.

The third part presents the application of the proposed framework to decision problems, showing how they can be cast as probabilistic continuous constraint spaces and using

methods proposed in part II to solve them. Comparisons are made with the classical techniques to solve this kind of problems.

Chapter 6 illustrates the application of the probabilistic continuous constraint framework to decision problems on non linear inverse problems. Inverse problems are defined and classical techniques to solve them presented, highlighting drawbacks of such approaches. The definition of an inverse problem as a probabilistic continuous constraint space is addressed and the capabilities of the PC framework are illustrated in three application problems. The first two show how to deal with non linear inverse problems, in general. The last, more complex, problem uses the framework on a real world ocean color application.

Chapter 7 illustrates the application of the probabilistic continuous constraint framework to reliability analysis problems. Reliability analysis problems are presented together with classical techniques to solve them, and some of their drawbacks discussed. The formulation of reliability analysis problems as probabilistic continuous constraint spaces is presented and the advantages of the PC framework are illustrated on a set of application problems.

Chapter 8 summarizes the work accomplished during this research and points out directions for future work.



## Part I

# Continuous Constraints and Uncertainty



## CHAPTER 2

# Continuous Constraint Programming

In the constraint programming paradigm, relations between variables are stated by means of constraints. By specifying the properties of solutions separately from their search, constraint programming is a form of declarative programming. In continuous constraint programming the domains of the variables are real intervals and the constraints are equations or inequalities represented by closed-form expressions. Consequently, interval analysis, which addresses the use of intervals in numerical computations, is an important component of continuous constraint programming. The relevant definitions and issues related to interval analysis are presented in section 2.1.

The main definitions regarding continuous constraint satisfaction problems (CCSP) (e.g. the variables of the problem, their interval domains and the constraints that relate them) as well as the continuous constraint reasoning approach to solve a CCSP are presented in section 2.2.

Section 2.3 subsequently overviews state of the art techniques related to continuous constraint reasoning, presenting alternative implementations for the general concepts of section 2.2, focusing on the approach adopted in the present thesis and introducing important concepts for the proposed framework.

## 2.1 Interval Analysis

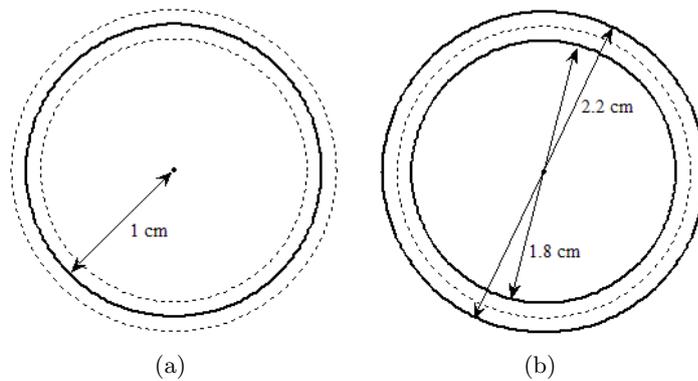
Interval analysis was introduced in the late 1950s [90] as a way to represent bounds in rounding and measurement errors, and to reason with these bounds in numerical

computations. Numerical methods that yield reliable results were developed to achieve this goal. In this section we address concepts and methods related to interval analysis, namely interval arithmetic, inclusion functions and interval methods.

### 2.1.1 Interval Arithmetic

Interval arithmetic is an extension of real arithmetic that allows numerical computations to be made with expressions where the operands assume interval values (i.e., a range of real values) instead of real values.

**Example 2.1.** Consider a circular object with a measured radius  $r$  of 1.0 cm. The observer is not certain about this measurement, and considers an error of 1 mm around it. The measured value thus ranges between 0.9 cm and 1.1 cm, i.e.,  $r \in [0.9, 1.1]$  cm, as illustrated in figure 2.1(a). Using interval arithmetic and the relation between diameter ( $d$ ) and radius ( $d = 2r$ ) it is  $d \in [2, 2] \times [0.9, 1.1] = [1.8, 2.2]$  cm, as illustrated in figure 2.1(b).



**Figure 2.1:** Model and propagate uncertainty. (a) Model uncertainty in the measurement of the radius. (b) Propagation of uncertainty to the calculus of the diameter.

This simple example illustrates the capabilities of intervals to represent uncertainty and interval arithmetic to guarantee rigorous results while reasoning with uncertain values.

**Definition 2.1 (Real Interval)** *A real interval is a closed connected set of real numbers.*

Non-empty intervals are represented by capital letters<sup>1</sup>,

$$X = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} : \underline{x} \leq x \leq \bar{x}\},$$

where  $\underline{x}$  ( $\bar{x}$ )  $\in \mathbb{R} \cup \{-\infty, +\infty\}$  is the interval lower (upper) bound and  $\underline{x} \leq \bar{x}$ .

The empty interval is represented by  $\emptyset$ .

An interval is bounded if both bounds are real numbers (i.e.  $\underline{x}, \bar{x} \notin \{-\infty, +\infty\}$ ). Otherwise it is unbounded.

An interval where the lower and upper bounds are equal (i.e.,  $\underline{x} = \bar{x} = x$ ) is called degenerated interval and is represented by  $[x, x]$  or only  $[x]$ .

As usual  $\mathbb{IR}$  denotes the set of all intervals over  $\mathbb{R}$ , defined as

$$\mathbb{IR} = \{[\underline{x}, \bar{x}] : \underline{x}, \bar{x} \in \mathbb{R} \cup \{-\infty, +\infty\}, \underline{x} \leq \bar{x}\} \cup \emptyset.$$

Informally, a real number is a value that represents a quantity along a continuum and can be thought of as a point on an infinitely long line. In general it requires an infinite decimal representation, such as  $\pi = 3.1415926535\dots$ , where the sequence continues indefinitely, possibly non-periodically. Since there are *infinitely* many, not all real numbers are machine representable, since machines are restricted to represent a finite set of elements.

To address this limitation, computers use a floating point system<sup>2</sup> for representing real numbers, whose elements are a finite subset,  $\mathbb{F}$ , of the reals, the  $\mathbb{F}$ -numbers. This subset includes the real number zero (0) as well as the infinity symbols  $-\infty$  and  $+\infty$  (which are not reals). It is totally ordered, following the ordering of  $\mathbb{R}$ . For any  $\mathbb{F}$ -number  $f \in \mathbb{F}$  we have that  $-\infty \leq f \leq +\infty$  and  $f^-$  ( $f^+$ ) is the  $\mathbb{F}$ -number immediately below (above)  $f$  in the total order. By definition,  $-\infty^- = -\infty$  and  $+\infty^+ = +\infty$ .

**Definition 2.2 ( $\mathbb{F}$ -Interval)** An  $\mathbb{F}$ -interval is the empty interval or an interval where both its bounds are  $\mathbb{F}$ -numbers.

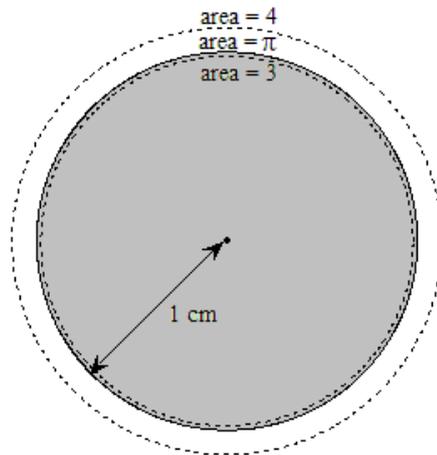
<sup>1</sup>Closed non-empty intervals will be considered, except when  $\pm\infty$  are in their bounds.

<sup>2</sup>The IEEE has standardized the computer representation for binary floating point numbers in IEEE 754 [66]. This standard is followed by almost all modern machines.

**Definition 2.3 (Canonical Interval)** *A canonical interval is either a degenerated interval or an interval where the lower and upper bounds are two consecutive  $\mathbb{F}$ -numbers.*

Since computers use floating point arithmetic, which relies on a finite set of elements, the result of numerical computations is often not the correct mathematical value, but an  $\mathbb{F}$ -number that approximates it. Although exact values may not be represented,  $\mathbb{F}$ -intervals can rigorously enclose a value that is not machine representable.

**Example 2.2.** Consider the area of a circle with radius  $r = 1.0$  cm, where  $area = \pi r^2$ . Since  $\pi$  is not machine representable, a computer may only achieve an approximation of this value. In fact, given an interval that binds the value of  $\pi$  it is possible to enclose the correct value of the area. For example, assuming that  $\pi \in [3, 4]$ , interval arithmetic guarantees that  $area \in [3, 4] \times [1, 1]^2 = [3, 4]$ , which encloses the exact value of the area ( $\pi$ ), as shown in figure 2.2.



**Figure 2.2:** Enclosure of the area exact value:  $3 \leq area = \pi \leq 4$ .

The previous example overcomes the insufficiencies of a floating point system by representing  $\pi$  with an interval (in this case a very crude one) and computing the area with interval arithmetic. Although the method does not provide the exact value for the area, it encloses its value with a given precision. Clearly, the narrower is the representation

of  $\pi$  the narrower is the computed interval representing the area. This method can be extended to any real number.

A classical example is the number 0.1 (a real number that is not an  $\mathbb{F}$ -number in binary representation). If the machine rounding mode is set downwards, then it is represented by the largest  $\mathbb{F}$ -number not greater than 0.1. If the rounding mode is set upwards, then it is represented by the smallest  $\mathbb{F}$ -number not lower than 0.1. Either way, whenever a mathematical computation is made with this number, the rounding error will be propagated to the result, possibly resulting in a crude approximation of the exact value.

Given a real number  $r \in \mathbb{R}$ , there are two  $\mathbb{F}$ -numbers,  $\nabla r$  and  $\Delta r \in \mathbb{F}$ , that are the *closest* machine representations of  $r$ .  $\nabla r$  is the largest  $\mathbb{F}$ -number not greater than  $r$ , whereas  $\Delta r$  is the smallest  $\mathbb{F}$ -number not lower than  $r$ . By definition  $\nabla -\infty = -\infty$ ,  $\Delta -\infty =$  smallest  $\mathbb{F}$ -number,  $\nabla +\infty =$  largest  $\mathbb{F}$ -number and  $\Delta +\infty = +\infty$ .

**Definition 2.4 ( $\mathbb{F}$ -Interval Approximation of a Real Interval)** *For any real interval  $X = [\underline{x}, \bar{x}]$ , there is a corresponding  $\mathbb{F}$ -interval, given by  $F_{\text{apx}}(X) = [\nabla \underline{x}, \Delta \bar{x}]$ , such that  $X \subseteq F_{\text{apx}}(X)$ .*

To acknowledge and consider rounding errors, instead of computing with the real number 0.1, interval arithmetic adopts the corresponding interval  $[\nabla 0.1, \Delta 0.1]$ .

Interval analysis is thus an important tool to bound rounding errors in computations due to machine limitations on their representation of real numbers.

In the following, the term interval will be used in a generic way to refer both to real intervals and to  $\mathbb{F}$ -intervals, unless otherwise stated.

Since intervals are sets, all set operations can be applied to them, namely, union ( $\cup$ ), intersection ( $\cap$ ) and inclusion ( $\subseteq$ ). While the intersection between two intervals is still an interval, this is not the case with the union of two disjoint intervals, where the result is a set that cannot be represented exactly by a single interval.

To address this situation another important binary operation, union hull, represented by the symbol  $\uplus$ , is defined as the smallest interval containing all the elements of its

arguments. It differs from normal union only when the arguments are disjoint. Given two intervals  $X = [\underline{x}, \bar{x}]$  and  $Y = [\underline{y}, \bar{y}]$ , then  $X \uplus Y = [\min(\underline{x}, \underline{y}), \max(\bar{x}, \bar{y})]$ .

Several useful functions over intervals will be used in this thesis. In particular, the functions that retrieve the lower and upper bounds and that compute the center and the width of a non-empty interval  $X = [\underline{x}, \bar{x}]$  are, respectively, denoted as:

$$\begin{aligned} \inf(X) &= \underline{x} \\ \sup(X) &= \bar{x} \\ \text{mid}(X) &= (\underline{x} + \bar{x})/2 \\ \text{wid}(X) &= \bar{x} - \underline{x} \end{aligned}$$

Functions  $\inf$ ,  $\sup$  and  $\text{mid}$  are undefined for an empty interval, whereas  $\text{wid}(\emptyset) = 0$ .

The generalization of intervals to several dimensions is of major relevance in this thesis.

**Definition 2.5 (Box)** *An  $n$ -dimensional box (or  $n$ -box)  $B$  is the Cartesian product of  $n$  intervals and is denoted by  $I_1 \times \dots \times I_n$ , where each  $I_i$  is an interval:*

$$B = \{ \langle d_1, \dots, d_n \rangle : d_1 \in I_1, \dots, d_n \in I_n \}$$

*If  $\exists_{1 \leq i \leq n} I_i = \emptyset$  then  $B$  is an empty box and is represented by  $\emptyset$ .*

*If  $\exists_{1 \leq i \leq n} I_i$  is degenerated then  $B$  is a degenerated box.*

*If  $\forall_{1 \leq i \leq n} I_i$  is bounded then  $B$  is a bounded box.*

Extending the interval notation, the set of all boxes over  $\mathbb{R}^n$  is denoted by  $\mathbb{I}\mathbb{R}^n$ .

**Definition 2.6 (F-Box)** *An  $\mathbb{F}$ -box is a box where all its dimensions are  $\mathbb{F}$ -intervals.*

The operations defined on intervals, namely, union, union hull, intersection and inclusion can be generalized to boxes. Intersection, union and union hull are obtained by applying the corresponding interval operation componentwise and, except for the union, the result is a box. For the inclusion operation the result is the conjunction of the componentwise interval inclusion operation.

Other useful functions over boxes define their width, volume and center. Given a non-empty box  $B = I_1 \times \cdots \times I_n$ :

$$\begin{aligned} \text{wid}(B) &= \max(\text{wid}(I_i)), \quad 1 \leq i \leq n \\ \text{vol}(B) &= \prod_{i=1}^n \text{wid}(I_i) \\ \text{mid}(B) &= \langle \text{mid}(I_1), \dots, \text{mid}(I_n) \rangle \end{aligned}$$

Contrary to functions  $\text{wid}$  and  $\text{vol}$ , function  $\text{mid}$  is undefined for an empty box.

Given a basic arithmetic operator (sum, difference, product or quotient) the corresponding interval arithmetic operator is an extension of the former to real intervals, both the operands and result being replaced by real intervals. The obtained interval is the set of all the values that result from a point-wise evaluation of the arithmetic operator on all the values of the operands. In practice these extensions simply consider the bounds of the operands to compute the bounds of the result, since the involved operations are monotonic.

**Definition 2.7 (Basic Interval Arithmetic Operators)** *Let  $X$  and  $Y$  be two bounded intervals. The basic arithmetic operators on intervals are defined as:*

$$X \diamond Y = \{x \diamond y : x \in X \wedge y \in Y\} \quad \text{with } \diamond \in \{+, -, \times, /\}$$

*Under the basic interval arithmetic,  $X/Y$  is undefined if  $0 \in Y$ .*

*Given two real intervals  $[\underline{x}, \bar{x}]$  and  $[\underline{y}, \bar{y}]$  these operations can be defined as:*

$$\begin{aligned} [\underline{x}, \bar{x}] + [\underline{y}, \bar{y}] &= [\underline{x} + \underline{y}, \bar{x} + \bar{y}] \\ [\underline{x}, \bar{x}] - [\underline{y}, \bar{y}] &= [\underline{x} - \bar{y}, \bar{x} - \underline{y}] \\ [\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}] &= [\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})] \\ [\underline{x}, \bar{x}] / [\underline{y}, \bar{y}] &= [\underline{x}, \bar{x}] \times [1/\bar{y}, 1/\underline{y}] \quad \text{if } 0 \notin [\underline{y}, \bar{y}] \end{aligned}$$

To extend these evaluations to  $\mathbb{F}$ -intervals we must consider the outward rounding of each bound to the *closest*  $\mathbb{F}$ -number. Given two  $\mathbb{F}$ -intervals,  $Z$  and  $W$ , and a real interval  $R$  defined by  $R = Z \diamond W$  with  $\diamond \in \{+, -, \times, /\}$ , then the  $\mathbb{F}$ -interval resulting from this operation is  $F_{\text{apx}}(R)$ .

The properties of basic operations for intervals present some remarkable differences from their properties in  $\mathbb{R}$ . Consider, for instance,  $X = [\underline{x}, \bar{x}]$  and the operation  $X - X$ . The result is equal to  $[\underline{x} - \bar{x}, \bar{x} - \underline{x}]$  and not  $[0]$ , as could be expected. This is because  $X - X$  is interpreted as  $\{x - y : x, y \in X\}$ , rather than  $\{x - x : x \in X\}$ . Thus, the subtraction does not take into account the dependency of the two occurrences of  $X$ .

Addition and multiplication remain associative and commutative, but multiplication is not distributive with respect to addition. Instead, in interval arithmetic the weaker subdistributivity property holds:  $X * (Y + Z) \subseteq X * Y + X * Z$ .

Several extensions to the basic interval arithmetic were proposed over the years and are available in extended interval arithmetic libraries [1, 21, 74, 107], namely a) redefinition of the division operator, allowing the denominator to contain zero [69, 104], b) generalization of interval arithmetic to unbounded interval arguments and results as union of disjoint intervals [62], and c) extension of the set of basic interval operators to other elementary functions (e.g., *exp*, *ln*, *power*, *sin*, *cos*).

### 2.1.2 Inclusion Functions

Since the properties of interval operations differ from their properties in  $\mathbb{R}$  it is not always possible to obtain sharp enclosures for the image of a real function by simply using interval arithmetic. In fact, different expressions for a real function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , which are mathematically equivalent, do not yield the same result when variables are replaced by intervals. Usually these results are much larger than the exact image of  $f$  over a box  $B$ .

For example, whereas the real expression  $x - x$  is equivalent to 0, its interval arithmetic evaluation for  $x \in [\underline{x}, \bar{x}]$  is not  $[0]$ , but rather the interval  $[\underline{x} - \bar{x}, \bar{x} - \underline{x}]$ , as discussed previously. In fact, this interval includes the correct value 0, but its width is twice the width of the original interval (much wider than the width of  $[0]$  - the exact interval for the expression over  $[\underline{x}, \bar{x}]$ ).

Consequently, in interval analysis special attention has been devoted to the definition of interval functions that compute sharp interval images of real functions [91].

The concepts presented in this section are adapted from [67] and we chose to use the more generic definition of inclusion function instead of that of interval extension

proposed by Moore [91]. In this context it is important to explicitly define the closely-related concepts of function and expression and distinguish between real and interval functions.

A function is a mapping from elements of a set, the domain, to another set, the codomain. The subset of the codomain restricted to the elements that are mapped by the function is the image of the function.

In a real function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , the elements of the domain are  $n$ -tuples of real values and the elements of the codomain are real values. The image of a real function  $f$  over a domain  $D$ ,  $\{f(x) : x \in D\}$  is denoted by  $f^*(D)$ .

A function may be represented by a closed-form expression and, as a direct consequence of arithmetic operators properties (e.g., distributive and commutative properties of the addition and multiplication real operators), different expressions may represent the same function.

In an interval function  $[f] : \mathbb{IR}^n \rightarrow \mathbb{IR}$ , the elements of the domain are  $n$ -boxes and the elements of the codomain are intervals.

**Definition 2.8 (Inclusion Function)** Consider a real function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . An interval function  $[f] : \mathbb{IR}^n \rightarrow \mathbb{IR}$  is an inclusion function of  $f$  iff:

$$\forall B \in \mathbb{IR}^n \quad f^*(B) \subseteq [f](B)$$

Therefore, any inclusion function  $[f]$  over a box  $B$ , produces an outer estimate of  $f^*(B)$ . In fact,  $[f]$  provides a sound evaluation of  $f$ , since the correct real value is not lost.

The following properties are usually considered for inclusion functions.

**Property 2.1 (Inclusion Functions Properties)** An inclusion function of  $f$ ,  $[f] : \mathbb{IR}^n \rightarrow \mathbb{IR}$ , is:

**Thin** if  $[f](x) = f(x)$ , for any point  $x \in \mathbb{R}^n$ .

**Convergent** if, for any sequence of boxes  $B_k \in \mathbb{IR}^n$ , with  $k \in \mathbb{N}$ ,

$$\lim_{k \rightarrow \infty} \text{wid}(B_k) = 0 \Rightarrow \lim_{k \rightarrow \infty} \text{wid}([f](B_k)) = 0.$$

**Optimal** if, for any  $B \in \mathbb{IR}^n$ ,  $[f](B) = f^*(B)$ .

**Inclusion Monotonic** if, for any boxes  $B_1, B_2 \in \mathbb{IR}^n$ ,

$$B_1 \subseteq B_2 \Rightarrow [f](B_1) \subseteq [f](B_2).$$

Notice that, if  $f$  is continuous, a convergent inclusion function is necessarily thin. The convergence of inclusion functions is required for proving the convergence of interval algorithms.

**Definition 2.9 (Order of Convergence)** Let  $[f]$  be an inclusion function of  $f$  over a box  $B \in \mathbb{IR}^n$ . The convergence order of  $[f]$  is  $o$  if there exists a positive constant  $k$  such that the inequality

$$\text{wid}([f](B')) - \text{wid}(f^*(B')) \leq k(\text{wid}(B'))^o$$

holds for every  $B' \subseteq B$ .

Obtaining inclusion functions is one of the most important problems that interval analysis deals with. Besides the natural inclusion functions, more advanced forms of inclusion functions have been proposed. Among those, the most popular include centered forms (e.g., Centered interval extension and Mean Value interval extension), introduced by Moore in [91] and are specially useful when the variables domains are sufficiently small. Taylor forms are higher degree generalizations of the centered forms and are based on the Taylor series expansion around a point, where the error term is bounded by an interval.

Both natural inclusion functions and Taylor inclusion functions are used in this thesis and described next.

### Natural Inclusion Functions

The natural inclusion function  $[f]_N$  of a real function  $f$  simply replaces real arithmetic by interval arithmetic, each variable by its interval domain and constants by degenerated intervals.

Natural inclusion functions are thin and inclusion monotonic. Moreover if  $f$  is continuous inside a box  $B$ , the natural inclusion function of  $f$  is optimal when each variable occurs only once in  $f$  [91]. When a variable appears several times in  $f$ , the evaluation by interval arithmetic generally produces an over-estimate of the optimal image enclosure, because the correlation between occurrences of the same variable is lost and two occurrences of the same variable are handled as independent variables. This is known as the dependency problem of interval arithmetic.

When computing an expression using interval arithmetic, the form of the expression can thus dramatically impact on the width of the resulting interval.

**Example 2.3.** Consider the two arithmetic expressions

$$(a + b)x \quad \text{and} \quad ax + bx$$

which are, of course, equivalent. However their natural extensions are not. For example, replacing the points,  $a$ ,  $b$  and  $x$ , by the intervals,  $A = [10]$ ,  $B = [-10]$  and  $X = [10, 20]$ , their natural extensions are:

$$\begin{aligned} (A + B)X &= [0] \times [10, 20] = [0] \\ AX + BX &= [100, 200] + [-200, -100] = [-100, 100] \end{aligned}$$

Evaluating the first expression produces an interval that is as narrow as possible, whereas evaluation of the second expression does not produce such a sharp result because it contains multiple occurrences of the interval variable  $X$ .

---

The natural inclusion function  $[f]_N$  for  $f$  is convergent inside any bounded box  $B$  where it is defined. Furthermore, if  $f$  is Lipschitz continuous inside  $B$ <sup>1</sup> then  $[f]_N$  has a linear order of convergence in  $B$  [96].

For a given real function the best inclusion functions, in the sense of minimal over-estimation, are those that minimize the dependency problem. Although it is often

---

<sup>1</sup>Intuitively, a Lipschitz continuous function is limited in how fast it can change. A line joining any two points on the graph of this function will never have a slope steeper than a certain number. A function  $f : B \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  that is differentiable in  $B$  is Lipschitz continuous if its first derivative is bounded.

impossible to construct inclusion functions without multiple occurrences of the same variable, inclusion functions with fewer repeated occurrences are usually preferable.

### Taylor Inclusion Functions

A Taylor model [84] is a kind of Taylor form that computes a high order polynomial approximation of a function through a multivariate Taylor expansion around a point, with a remainder term that rigorously bounds the approximation error.

To define multivariate Taylor models and inclusion functions it is useful to adopt multi-index notation, which simplifies multivariate generalizations of more familiar univariate definitions.

A multi-index  $\alpha = \langle \alpha_1, \dots, \alpha_k \rangle$  is a  $k$ -tuple of non-negative integers. For a multi-index  $\alpha$  and a tuple  $x = \langle x_1, \dots, x_k \rangle \in \mathbb{R}^k$ , the following operations can be defined.

- **Norm:**  $|\alpha| = \alpha_1 + \dots + \alpha_k$ .
- **Factorial:**  $\alpha! = \prod_{i=1}^k \alpha_i!$ .
- **Power:**  $x^\alpha = \prod_{i=1}^k x_i^{\alpha_i}$ .
- **Higher order derivative:**  $\frac{\partial^\alpha}{\partial x^\alpha} = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_k^{\alpha_k}}$

**Example 2.4.** Let  $\alpha = \langle 2, 3, 1 \rangle$  and  $x = \langle 4, -1, 2 \rangle$ . Then

- $|\alpha| = 2 + 3 + 1 = 6$
- $\alpha! = 2! \times 3! \times 1! = 12$
- $x^\alpha = 4^2 \times (-1)^3 \times 2^1 = -32$
- $\frac{\partial^\alpha}{\partial x^\alpha} = \frac{\partial^6}{\partial x_1^2 \partial x_2^3 \partial x_3^1}$

---

**Definition 2.10 (Taylor Model)** *Given a function  $f : \mathbb{R}^k \rightarrow \mathbb{R}$ ,  $n + 1$  times continuously differentiable in a box  $B \subseteq \mathbb{R}^k$  and  $\tilde{x}$  a point in  $B$ , then the  $n^{\text{th}}$  order Taylor model of  $f$  around  $\tilde{x}$  in  $B$  is a pair  $\langle p, R \rangle$ , where  $p$  is a polynomial and  $R$  an interval, satisfying  $f(x) \in p(x) + R$ , for all  $x \in B$ .*

A common way of building a Taylor model adopts a multivariate Taylor expansion of  $f$ . Using multi-index notation, the  $n^{\text{th}}$  order Taylor expansion of  $f$  in  $B$ , expanded at  $\tilde{x} \in B$ , is:

$$f(x) = f(\tilde{x}) + \sum_{|\alpha|=1}^n \frac{1}{\alpha!} \frac{\partial^\alpha f(\tilde{x})}{\partial x^\alpha} (x - \tilde{x})^\alpha + \sum_{|\alpha|=n+1} r_\alpha(\xi) (x - \tilde{x})^\alpha$$

where  $\xi$  lies between  $x$  and  $\tilde{x}$  and  $r_\alpha(\xi)$  is defined as

$$r_\alpha(\xi) = \frac{1}{\alpha!} \frac{\partial^\alpha f(\xi)}{\partial x^\alpha}$$

Since  $x, \tilde{x} \in B$ , then  $\xi$  is also in  $B$ . Therefore, given an inclusion function  $\left[ \frac{\partial^\alpha f}{\partial x^\alpha} \right](B)$  of the partial derivative  $\frac{\partial^\alpha f(\xi)}{\partial x^\alpha}$ , the remainder is bounded by the inclusion function:

$$r_\alpha(\xi) \in [r_\alpha](B) = \frac{1}{\alpha!} \left[ \frac{\partial^\alpha f}{\partial x^\alpha} \right](B)$$

The above Taylor expansion yields  $\langle p, R \rangle$ , a Taylor model of  $f$  in  $B$  where:

$$p(x) = f(\tilde{x}) + \sum_{|\alpha|=1}^n \frac{1}{\alpha!} \frac{\partial^\alpha f(\tilde{x})}{\partial x^\alpha} (x - \tilde{x})^\alpha \quad (2.1)$$

and

$$R = \sum_{|\alpha|=n+1} [r_\alpha](B) (B - \tilde{x})^\alpha \quad (2.2)$$

The Taylor inclusion function for  $f$ , denoted as  $[f]_T$ , is based on the Taylor models introduced in [16].

**Definition 2.11 (Taylor Inclusion Function)** *Given the Taylor model  $\langle p, R \rangle$  of  $f$  in  $B$ , with  $p$  defined as in (2.1) and  $R$  defined as in (2.2), then a Taylor inclusion function for  $f$  in  $B$  (of the same order as the Taylor model) is given by  $[f]_T(B) = [p](B) + R$ .*

If  $[p](B)$  is an optimal inclusion function then the  $n^{\text{th}}$ -order Taylor inclusion function has order of convergence  $(n + 1)$  as shown in [83].

Nevertheless, in practice, the computation of an optimal inclusion function for  $p(B)$  is a difficult problem, since it requires the computation of the image of an  $n^{\text{th}}$  degree polynomial. In [83], very effective approximation methods are presented that maintain the order of convergence of  $n + 1$ .

### 2.1.3 Interval Methods

Interval methods for finding roots of equations with one variable are frequently used in constraint programming due to their efficiency and reliability. In particular the interval Newton method combines the classical Newton method, the mean value theorem, and interval analysis. The result is an iterative method that can be used to prove non-existence of solutions of a nonlinear equation or to provide rigorous bounds for the solutions (eventually proving existence and uniqueness of such solutions). Such capabilities can be used either in isolation to provide an enclosure for the zeros of a real function, or as part of branch and prune algorithms to provide rigorous bounds for particular solutions.

#### Univariate Interval Newton Method

Given a function  $f : \mathbb{R} \rightarrow \mathbb{R}$  continuous and differentiable, and an interval  $X$ , the univariate interval Newton method computes an enclosure for the set  $\{x \in X : f(x) = 0\}$  as follows.

The well known mean value theorem can be formulated as:

$$\forall_{x_1, x_2 \in X} \exists_{\xi \in X} f(x_1) - f(x_2) = (x_1 - x_2)f'(\xi) \quad (2.3)$$

Assuming that there exists  $x^* \in X$  such that  $f(x^*) = 0$  and  $\tilde{x} \in X$ , making  $x_1 = x^*$  and  $x_2 = \tilde{x}$  in (2.3)

$$\exists_{\xi \in X} 0 = f(x^*) = f(\tilde{x}) + f'(\xi)(x^* - \tilde{x})$$

and hence  $x^* = \tilde{x} - \frac{f(\tilde{x})}{f'(\xi)}$  for some  $\xi \in X$ . If  $[f'](X)$  is an inclusion function of the

derivative of  $f$  over  $X$ , then we can replace  $f'(\xi)$  for all its possible values in  $X$

$$x^* \in \tilde{x} - \frac{f(\tilde{x})}{[f'](X)} \quad \text{for any } \tilde{x} \in X \quad (2.4)$$

The previous equation forms the basis of the univariate interval Newton operator:

$$N(X) = \tilde{x} - \frac{f(\tilde{x})}{[f'](X)} \quad (2.5)$$

Because of (2.4), any solutions of  $f(x) = 0$  that are in  $X$  must also be in  $N(X)$ . Furthermore  $N(X) \subset X$  implies that there is a unique solution of  $f(x) = 0$  within  $N(X)$ , and hence within  $X$  [92, 98].

Therefore, given a continuous and differentiable function  $f(x)$ , the interval version of the Newton's method computes an enclosure of a zero  $x^*$  in an interval  $X$  by iterating

$$X_{k+1} = N(X_k) \cap X_k, \quad (2.6)$$

where  $X_0 = X$ .

The interval Newton iteration (2.6) may achieve a quadratic order of convergence [57], i.e., the width of  $X_{k+1}$  at every step is up to a constant factor, less than the square of the width at the previous step,  $X_k$ . However, the quadratic order is only achieved when the iteration is contracting, i.e., when  $N(X_k) \subseteq X_k$ . When this condition is not fulfilled, the progression can be slow.

In the above definitions  $f$  has been assumed to be a real function. However, the definition can be naturally extended to deal with interval functions that include parametric constants represented by intervals [58]. In this case the intended meaning is to represent the family of real functions defined by any possible real valued instantiation for the interval constants. The existence of a root means that there is a real valued combination, among the variable and all the interval constants, that zeros the function.

### Multivariate Interval Newton Methods

Multivariate interval Newton methods (interval Newton method for multivariate functions) are specially suited for finding roots of systems of  $n$  equation with  $n$  variables.

They are more complex to implement than their univariate counterpart since they require the computation of the inverse of the interval Jacobian matrix. The use of different methods to compute this matrix distinguish the different multivariate interval Newton methods (e.g. [59, 75, 91]).

Multivariate methods share the properties presented for the univariate method, with the exception of the quadratic convergence property that is only verified in specific conditions. These methods can be effectively applied in cases where the search space is a small box enclosing a single root either to isolate the root or to prove its existence and uniqueness. However, for large boxes, the narrowing achieved by these methods does not justify, in general, the computational costs of their implementation.

## 2.2 Continuous Constraint Satisfaction Problems

Continuous Constraint Satisfaction Problems (hereafter referred as CCSPs) are mathematical problems defined over a set of variables ranging over real intervals. In this section we present the basic notions of constraint satisfaction methods for CCSPs.

**Definition 2.12 (Domain of a Variable)** *The domain of a variable is the set of values that the variable can assume. In CCSPs, the domains of the variables are real intervals.*

Constraints further reduce the values from the domains of its variables which are acceptable.

**Definition 2.13 (Numerical Constraint)** *A numerical constraint  $c$  is a pair  $(s, \rho)$ , where  $s$  (the constraint scope) is a tuple of  $n$  variables  $\langle x_1, \dots, x_n \rangle$  and  $\rho$  (the constraint relation) is a subset of the box  $B = I_1 \times \dots \times I_n$ ,  $\rho \subseteq B$  (where  $I_i$  is the domain of variable  $x_i$ ).*

*We consider constraint relations of the form  $\rho = \{d \in B \cap D_f : f(d) \diamond 0\}$ , where  $f : D_f \rightarrow \mathbb{R}$  is a function represented by a closed-form expression and  $\diamond \in \{=, \leq\}$ .*

A CCSP is defined by a set of numerical constraints, together with the variables used in those constraints and their domains.

**Definition 2.14 (Continuous Constraint Satisfaction Problem)** *A CCSP is a triple  $\langle X, D, C \rangle$  where  $X$  is a tuple of  $n$  real variables  $\langle x_1, \dots, x_n \rangle$ ,  $D = I_1 \times \dots \times I_n$  is a bounded box, where each real interval  $I_i$  is the domain of variable  $x_i$ , and  $C$  is a finite set of numerical constraints on (subsets of) the variables in  $X$ .*

The Cartesian product of the variable domains,  $D$ , is the initial domain or initial search space of the CCSP.

Since different constraints in a CCSP may have different scopes, these can be obtained by projection.

**Definition 2.15 (Tuple Projection)** *Let  $\langle x_1, \dots, x_n \rangle$  be an  $n$ -tuple of variables and  $d = \langle d_1, \dots, d_n \rangle$  a corresponding  $n$ -tuple of values. Let  $s = \langle x_{i_1}, \dots, x_{i_m} \rangle$  be a tuple of  $m$  variables where  $1 \leq i_j \leq n$ . The tuple projection of  $d$  wrt  $s$  is:*

$$d[s] = \langle d_{i_1}, \dots, d_{i_m} \rangle$$

*For simplicity, when  $s$  is a tuple with a single variable the brackets are ignored both in  $s$  and in the result of the projection (i.e., the result is a real value).*

Tuple projection can be extended to box projection to deal with the domains of the variables in a CCSP.

**Definition 2.16 (Box Projection)** *Let  $\langle x_1, \dots, x_n \rangle$  be an  $n$ -tuple of variables and  $B = I_1 \times \dots \times I_n$  a corresponding box. Let  $idx = \langle i_1, \dots, i_j \rangle$  be an  $m$ -tuple of indices, where  $1 \leq i_j \leq n$ ,  $m \leq n$ , whose corresponding tuple of variables is  $s = \langle x_{i_1}, \dots, x_{i_m} \rangle$ . The box projection of  $B$  wrt  $idx$  is the  $m$ -dimensional box:*

$$\Pi_{idx}(B) = \{d[s] : d \in B\}$$

*For simplicity, when  $idx$  is a tuple with a single index the brackets may be ignored.*

**Example 2.5.** Consider the tuple of variables  $\langle x_1, x_2, x_3, x_4 \rangle$  and a box  $B = [21.1, 25.5] \times [2.1, 6.2] \times [11.5, 19.3] \times [-3.1, -2.6]$ . For  $s = \langle x_2, x_4 \rangle$  it is  $idx = \langle 2, 4 \rangle$  and  $\Pi_{\langle 2,4 \rangle}(B) = [2.1, 6.2] \times [-3.1, -2.6]$ .

---

A tuple of values satisfies a constraint if its projection onto the constraint scope is in the constraint relation. Hence, the definition of solution of a CCSP.

**Definition 2.17 (Solution)** *A solution of the CCSP  $\langle X, D, C \rangle$  is a tuple  $d \in D$  that satisfies all the constraints in  $C$ :*

$$\forall_{(s,\rho) \in C} d[s] \in \rho$$

Whereas in some CCSPs it is important to determine individual solutions, in many practical situations, due to the continuous nature of such solutions, the ultimate goal is to characterize the complete set of solutions.

**Definition 2.18 (Feasible Space)** *The feasible space of the CCSP  $P = \langle X, D, C \rangle$  is the set  $\mathcal{F}(P) \subseteq D$  of all solutions of the CCSP, defined as:*

$$\mathcal{F}(P) = \{d \in D : \forall_{(s,\rho) \in C} d[s] \in \rho\}$$

*When there is no possible ambiguity  $\mathcal{F}$  is used to denote the feasible space of a CCSP.*

Constraint reasoning aims at eliminating values from the initial search space that do not satisfy the constraints, by pruning and subdividing the search space until a stopping criterion is satisfied. Pruning is accomplished by eliminating sets of values that can be proved inconsistent.

**Definition 2.19 (Consistency)** *A set  $D' \subseteq D$  is consistent with CCSP  $\langle X, D, C \rangle$  iff it contains at least one solution (otherwise it is inconsistent):*

$$\exists_{d \in D'} \forall_{(s,\rho) \in C} d[s] \in \rho$$

To eliminate value combinations incompatible with a particular constraint, safe narrowing operators (mappings between sets) are associated with the constraint. These operators must be correct (do not eliminate solutions) and contracting (the obtained set is contained in the original).

**Definition 2.20 (Narrowing Operator)** *Let  $\langle X, D, C \rangle$  be a CCSP. An operator  $\mathcal{N} : 2^D \rightarrow 2^D$ , that defines a mapping between subsets of  $D$ , is a narrowing operator associated with a constraint  $(s, \rho) \in C$  iff:*

$$\forall_{D' \subseteq D}, \begin{cases} \mathcal{N}(D') \subseteq D' & (\text{contractance}) \\ \forall_{d \in D'} d \notin \mathcal{N}(D') \Rightarrow d[s] \notin \rho & (\text{correctness}) \end{cases}$$

The previous definition can be easily extended to consider a narrowing operator associated with more than one constraint (e.g. a narrowing operator derived from the multivariate interval Newton method). Nevertheless, for simplicity, we only consider narrowing operators associated with a single constraint.

The following properties are usually considered for narrowing operators.

**Property 2.2 (Properties of Narrowing Operators)** *Let  $\langle X, D, C \rangle$  be a CCSP and  $D \subseteq \mathbb{R}^n$ . A narrowing operator  $\mathcal{N} : 2^D \rightarrow 2^D$  is:*

**Inclusion Monotonic** *if, for any  $D_1, D_2 \subseteq D$ ,  $D_1 \subseteq D_2 \Rightarrow \mathcal{N}(D_1) \subseteq \mathcal{N}(D_2)$ .*

**Idempotent** *if, for any  $D' \subseteq D$ ,  $\mathcal{N}(\mathcal{N}(D')) = \mathcal{N}(D')$ .*

*A set  $D' \subseteq D$  is a **fixed point** of  $\mathcal{N}$  iff  $\mathcal{N}(D') = D'$ .*

Once narrowing operators are associated with the constraints of the CCSP, the pruning of variables domains can be achieved through constraint propagation. Narrowing operators associated with a constraint eliminate some incompatible values from the domain of its variables and this information is propagated to all constraints with common variables in their scopes. The process terminates when a fixed point is reached i.e., the domains can not be further reduced by any narrowing operator.

**Definition 2.21 (Constraint Propagation Algorithm)** Let  $\langle X, D, C \rangle$  be a CCSP and  $\mathcal{N}_s$  be a set of narrowing operators associated with the constraints in  $C$ . A constraint propagation algorithm  $\mathcal{CPA}$  defines a mapping between subsets of  $D$  where:

$$\forall D' \subseteq D, \begin{cases} \mathcal{CPA}(D', \mathcal{N}_s) \subseteq D' & (\text{contractance}) \\ \forall d \in D', d \notin \mathcal{CPA}(D', \mathcal{N}_s) \Rightarrow \exists (s, \rho) \in C, d[s] \notin \rho & (\text{correctness}) \\ \forall \mathcal{N} \in \mathcal{N}_s, \mathcal{N}(\mathcal{CPA}(D', \mathcal{N}_s)) = \mathcal{CPA}(D', \mathcal{N}_s) & (\text{fixed point}) \end{cases}$$

The pruning achieved through constraint propagation is highly dependent on the ability of the narrowing operators for discarding inconsistent value combinations [35]. Further pruning is usually obtained by splitting the domains and reapplying constraint propagation to each sub-domain. In general, continuous constraint reasoning is based on such a branch and prune process which will eventually terminate due to the imposition of conditions on the branching process (e.g. *small enough* domains are not considered for branching).

Remarkably, since no solution is lost during the process, constraint reasoning provides a safe method for computing an enclosure of the feasible space of a CCSP.

In this thesis we are interested in computing two set representations of the feasible space of a CCSP  $\langle X, D, C \rangle$ : one which includes it and another which is included in it. Constraint reasoning provides safe methods for computing these set representations, as described in detail in the next section.

## 2.3 Computing Feasible Space Approximations

It is usually impossible to exactly compute the feasible space of a CCSP, namely when it corresponds to a region with a non-linear boundary. For this reason, most constraint programming techniques rely on specific sets - boxes or unions of boxes - to represent such feasible space safely, although (possibly) not exactly.

This section presents state of the art techniques that implement such constraint reasoning. In subsection 2.3.1 the constraint propagation algorithm is presented together

with standard techniques for filtering the variables domains. Subsection 2.3.3 addresses boxes and feasible space representations, as well as the algorithms to compute them.

Since we are dealing with the computational aspects of constraint reasoning, in the following we consider  $\mathbb{F}$ -boxes.

### 2.3.1 Constraint Propagation

The constraint propagation algorithm for continuous domains is an adaptation of the original propagation algorithm AC3 [89] for finite domains. It consecutively applies each of the narrowing operators, associated with the constraints, to a given box. In each step, if the domain of a variable is reduced, this information is propagated to all other narrowing operators for which the current box is no longer a fixed point. Propagation terminates when the obtained box is a fixed point for all the narrowing operators, or, more realistically, when the boxes are considered sufficiently small.

It can be proved [100] that the propagation algorithm is correct and terminates independently from the order of application of the narrowing operators during the process. Moreover, if the narrowing operators are inclusion monotonic then the propagation algorithm is confluent (the result is independent from their order of application) and converges to the greatest common fixed point included in the initial search space [67].

In the following we assume that a narrowing operator  $\mathcal{N}$  associated with constraint  $f \diamond 0$ , discards a box  $B$  whenever the evaluation of the inclusion function  $[f]$  over  $B$  results on an interval that can not satisfy the constraint:

- when  $\diamond$  is  $=$  then  $\forall_{B \subseteq D} 0 \notin [f](B) \Rightarrow \mathcal{N}(B) = \emptyset$ ;
- when  $\diamond$  is  $\leq$  then  $\forall_{B \subseteq D} 0 \leq \inf([f](B)) \Rightarrow \mathcal{N}(B) = \emptyset$ .

Algorithm 1 is a pseudo-code description of the constraint propagation algorithm. It starts with a box  $B$  and the set of constraints of a CCSP. The algorithm maintains two sets of narrowing operators:  $\mathcal{N}s$  contains the narrowing operators for which  $B$  is not guaranteed to be a fixed point;  $S$  contains the narrowing operators for which  $B$  is a fixed point. Initially set  $S$  is empty and  $\mathcal{N}s$  contains all narrowing operators associated with the constraints (line 1). As the narrowing operators of  $\mathcal{N}s$  are applied to  $B$  (line 4), they are transferred to  $S$  if  $B$  becomes a fixed point of that operator (lines 6 – 8). The opposite may also happen to all the narrowing operators in  $S$  for

**Algorithm 1:**  $\mathcal{CPA}(B, C)$ 

```

Input:  $B$ : box;  $C$ : set of constraints;
Output:  $B$ : box;
1  $S \leftarrow \emptyset$ ;  $\mathcal{N}s \leftarrow \text{narrowingOps}(C)$ ;
2 while ( $\mathcal{N}s \neq \emptyset$ ) do
3    $\mathcal{N} \leftarrow \text{choose}(\mathcal{N}s)$ ;
4    $B' \leftarrow \mathcal{N}(B)$ ;
5   if  $B' = \emptyset$  then return  $\emptyset$ ;
6   if  $B' = B$  then
7      $\mathcal{N}s \leftarrow \mathcal{N}s \setminus \{\mathcal{N}\}$ ;
8      $S \leftarrow S \cup \{\mathcal{N}\}$ ;
9   else
10     $P \leftarrow \{\mathcal{N}' \in S : \exists x_i \in \text{vars}(\mathcal{N}') \Pi_i(B) \neq \Pi_i(B')\}$ ;
11     $\mathcal{N}s \leftarrow \mathcal{N}s \cup P$ ;
12     $S \leftarrow S \setminus P$ ;
13  end
14   $B \leftarrow B'$ 
15 end
16 return  $B$ ;

```

which the narrowed box  $B'$  is no longer guaranteed to be a fixed point (lines 9 – 13). A narrowing operator  $\mathcal{N}'$  is no longer guaranteed to be a fixed point of the current box if some interval regarding a variable in the constraint scope was narrowed. The loop stops when no more narrowing operators can be applied (line 2) and returns the current box (line 16). Whenever the application of a narrowing operator  $\mathcal{N}$  to a box  $B$  results in an empty box, the execution terminates by returning  $\emptyset$  (line 5), guaranteeing the correctness of the  $\mathcal{CPA}$ .

### 2.3.2 Consistencies

The fixed point obtained through constraint propagation characterizes a local consistency among the variables of the problem, which depends on the narrowing operators associated with each constraint (local) and the value combinations that are not pruned by them (consistent).

The most common local consistencies used in CCSPs, hull-consistency [81] (or 2B-consistency) and box-consistency [15, 36] (or some variation of them), are based on arc-consistency [89], extensively used in finite domains. Arc-consistency eliminates a

value from a variable domain if there is no support for this value in the domains of the other constraint variables.

### Hull Consistency

Hull-consistency guarantees arc-consistency only at the bounds of the variable domains. Intuitively, a constraint is hull-consistent with respect to a box, if there exists a solution of the constraint in every *face* of the box.

**Definition 2.22 (Hull-Consistency)** *Let  $c = \langle s, \rho \rangle$  be a numerical constraint and  $x_k \in s$  a variable. Let  $B$  be a box, where  $\Pi_k(B) = [\underline{x}_k, \bar{x}_k]$  is an  $\mathbb{F}$ -interval.*

*$\langle c, x_k \rangle$  is hull-consistent wrt  $B$  iff there are two points,  $p_1, p_2 \in B$  such that:*

$$\begin{array}{ll} p_1[s], p_2[s] \in \rho & (p_1 \text{ and } p_2 \text{ satisfy } c) \\ p_1[\langle x \rangle] \in [\underline{x}_k, \underline{x}_k^+) & (p_1 \text{ is in the box face with smallest } x \text{ value}) \\ p_2[\langle x \rangle] \in (\bar{x}_k^-, \bar{x}_k] & (p_2 \text{ is in the box face with largest } x \text{ value}). \end{array}$$

*Constraint  $c$  is hull-consistent wrt  $B$  iff, for every variable  $x_i \in s$ ,  $\langle c, x_i \rangle$  is hull-consistent wrt  $B$ .*

Algorithm *HC3* [15, 34] enforces hull-consistency on a set of primitive constraints obtained from the decomposition of the original constraints. Since it deals with primitive constraints it takes advantage of this simplified form to *invert* the constraints with respect to each of its variables. It then replaces the other variables by its interval domains and evaluates the resulting expression using interval arithmetic. The intervals obtained for each of the constraint variables are then intersected with their original domains.

Algorithm *HC4* [14] produces similar results but avoids explicit decomposition of a complex constraint, maintaining a tree representation of the original constraint. More recent algorithms [115] replace the tree by a representation of constraints with direct acyclic graphs (DAGs), thus allowing common sub-expressions to be shared and enhancing the constraint propagation process.

## Box Consistency

Box-consistency guarantees hull-consistency on unary projections of the constraints. Roughly speaking it consists of replacing all but one variable by its interval domain in the definition of hull-consistency.

**Definition 2.23 (Box-Consistency)** *Let  $c = \langle s, \rho \rangle$  be a numerical constraint, with  $f(x_1, \dots, x_n) \diamond 0$  (see definition 2.13) and  $x_k \in s$  a variable. Let  $B$  be a box where  $\Pi_{\langle 1, \dots, n \rangle}(B) = I_1 \times \dots \times I_{k-1} \times [\underline{x}_k, \bar{x}_k] \times I_{k+1} \times \dots \times I_n$  is an  $\mathbb{F}$ -box.*

*$\langle c, x_k \rangle$  is box-consistent wrt  $B$  (parameterized by the inclusion function  $[f]$ ) iff:*

$$\begin{aligned} \exists r_1 \in [f](I_1, \dots, I_{k-1}, [\underline{x}_k, \underline{x}_k^+], I_{k+1}, \dots, I_n) \quad r_1 \diamond 0 \\ \exists r_2 \in [f](I_1, \dots, I_{k-1}, [\bar{x}_k^-, \bar{x}_k], I_{k+1}, \dots, I_n) \quad r_2 \diamond 0 \end{aligned}$$

*Constraint  $c$  is box-consistent wrt  $B$  (parameterized by the inclusion function  $[f]$ ) iff, for every variable  $x_i \in s$ ,  $\langle c, x_i \rangle$  is box-consistent wrt  $B$ .*

Algorithm *BC3* [61] enforces box-consistency by combining binary search with the interval Newton method [91] to isolate the leftmost and rightmost zeros of the resulting system of univariate equations. Efficient enforcing algorithms result from the inclusion of adaptive shaving processes [49].

**Definition 2.24 (Consistent CCSP)** *A CCSP  $\langle X, D, C \rangle$  is (hull or box) consistent wrt  $D' \subseteq D$  iff every constraint  $c \in C$  is (respectively hull or box) consistent wrt  $D'$ .*

More sophisticated consistency techniques combine the above algorithms and extend them based on the structure of the constraints. Algorithm *BC4* [14] applies *HC3* to variables with a single occurrence in a constraint, and uses *BC3* otherwise, minimizing the dependency problem [91]. Other algorithms exploit the monotonicity properties of the constraints [9] and show that hull-consistency can be enforced in polynomial time if the constraint functions are all monotonic [29]. Algorithm *I-CSE* [7] exploits common sub-expressions obtaining a DAG that is rewritten into a new optimized system of constraints that can be processed by the tree-based algorithms.

Several consistency techniques rely on the combination of constraints to improve the precision of domain reductions.  $kB$ -consistency [81] and Box- $k$ -consistency [8] are generalizations of hull and box-consistency that enforce consistency properties on the overall constraint set. Algorithms Box- $k$  [8] and  $IBB$  [97] were proposed for handling together subsystems of  $k$  constraints from the original constraint set. Some algorithms [50], restricted to equation constraints, apply variants of the multivariate interval Newton method [91] that operate on the whole system of equations. Other algorithms [113], are based on constructive interval disjunction relying on the enforcement of other consistencies on slices of the current box and posterior assemblage of the obtained sub-boxes.

### 2.3.3 Constraint Reasoning

This subsection is concerned with specific needs of the probabilistic constraint framework proposed in this thesis and discusses how constraint reasoning computes two set representations of the feasible space of a CCSP.

Since the constraint satisfaction methods used in this work reason over boxes we will start by defining some relations between them.

**Definition 2.25 (Almost Disjoint Boxes)** *Two boxes  $A$  and  $B$  are almost disjoint iff  $\text{vol}(A \cap B) = 0$ .*

This definition can be extended to a set of boxes, as presented below.

**Definition 2.26 (Mutually Almost Disjoint Set of Boxes)** *A set of boxes  $\{B_1, \dots, B_n\}$  is mutually almost disjoint (or almost disjoint, for simplicity) iff  $\forall_{i \neq j \in \{1, \dots, n\}} B_i$  and  $B_j$  are almost disjoint.*

For simplicity, when the context allows it, we will refer to the feasible space of a CCSP simply as CCSP.

Constraint reasoning applies, repeatedly, branch and prune steps to reshape the initial search space (a box) maintaining a set of working boxes during the process, characterized as follows.

**Definition 2.27 (Outer Box Cover)** Let  $\langle X, D, C \rangle$  be a CCSP. The almost disjoint set of boxes  $\{B_1, \dots, B_n\}$ , where  $\forall_{1 \leq i \leq n} (B_i \subseteq D \wedge \text{vol}(B_i) > 0)$ , is an outer box cover of  $\mathcal{F}$  iff

$$\mathcal{F} \subseteq \bigcup_{i=1}^n B_i.$$

An outer box cover of  $\mathcal{F}$  is denoted by  $\mathcal{F}_\square$ . The union of its boxes is an outer approximation of  $\mathcal{F}$  and is denoted by  $\mathcal{F}^+$ .

A complementary concept is that of inner box cover. An inner box of a CCSP is a box totally contained in the feasible space, i.e., a box where all its points are solutions of the CCSP.

**Definition 2.28 (Inner Box)** Given a CCSP  $\langle X, D, C \rangle$ , a box  $B \subseteq D$  with  $\text{vol}(B) > 0$  is an inner box wrt  $\mathcal{F}$  iff  $B \subseteq \mathcal{F}$ .

There are techniques that identify inner boxes [67]. When the feasible space is defined by inequality constraints, one such simple technique relies on natural inclusion functions of the functions induced from the constraints relations, replacing the variables by the intervals of the box, and checking whether all values in the resulting interval are solutions for the constraints. When the feasible space is defined by equation constraints there are no inner boxes (see appendix A for an implementation).

**Definition 2.29 (Inner Box Cover)** Let  $\langle X, D, C \rangle$  be a CCSP. The almost disjoint set of boxes  $\{B_1, \dots, B_n\}$ , where  $\forall_{1 \leq i \leq n} (B_i \subseteq D \wedge \text{vol}(B_i) > 0)$ , is an inner box cover of  $\mathcal{F}$  iff

$$\bigcup_{i=1}^n B_i \subseteq \mathcal{F}.$$

An inner box cover of  $\mathcal{F}$  is denoted by  $\mathcal{F}_\blacksquare$ . The union of its boxes is an inner approximation of  $\mathcal{F}$  and is denoted by  $\mathcal{F}^-$ .

We are particularly interested in maintaining an inner box cover that is a subset of the outer box cover, hence, the notion of joint box cover.

**Definition 2.30 (Joint Box Cover)** *Let  $\langle X, D, C \rangle$  be a CCSP. A joint box cover of  $\mathcal{F}$ , be denoted by  $\mathcal{F}_{\boxplus}$ , is a pair  $\langle \mathcal{F}_{\square}, \mathcal{F}_{\blacksquare} \rangle$ , where  $\mathcal{F}_{\blacksquare} \subseteq \mathcal{F}_{\square}$ .*

The boxes of a joint box cover of a CCSP that are not in the inner box cover are called boundary boxes.

**Definition 2.31 (Boundary Box)** *Given a CCSP  $\langle X, D, C \rangle$  and a joint box cover of  $\mathcal{F}$ ,  $\mathcal{F}_{\boxplus} = \langle \mathcal{F}_{\square}, \mathcal{F}_{\blacksquare} \rangle$ ,  $B$  is a boundary box with respect to  $\mathcal{F}_{\boxplus}$  iff  $B \in \mathcal{F}_{\square} \setminus \mathcal{F}_{\blacksquare}$ .*

**Definition 2.32 (Boundary Box Cover)** *Given a CCSP  $\langle X, D, C \rangle$  and a joint box cover of  $\mathcal{F}$ ,  $\mathcal{F}_{\boxplus} = \langle \mathcal{F}_{\square}, \mathcal{F}_{\blacksquare} \rangle$ , the boundary box cover is  $\mathcal{F}_{\square} \setminus \mathcal{F}_{\blacksquare}$ .*

*The union of its boxes is a boundary approximation and is denoted by  $\Delta\mathcal{F}$ .*

Often, it is important to know whether a joint box cover represents the feasible space more accurately than another. The tighter relation ( $\preceq$ ) serves this purpose.

**Definition 2.33 (Tighter Joint Box Cover)** *Given a CCSP  $\langle X, D, C \rangle$  and two joint box covers of  $\mathcal{F}$ ,  $\mathcal{F}_{\boxplus_1} = \langle \mathcal{F}_{\square_1}, \mathcal{F}_{\blacksquare_1} \rangle$  and  $\mathcal{F}_{\boxplus_2} = \langle \mathcal{F}_{\square_2}, \mathcal{F}_{\blacksquare_2} \rangle$ ,  $\mathcal{F}_{\boxplus_2}$  is tighter than  $\mathcal{F}_{\boxplus_1}$ , (written  $\mathcal{F}_{\boxplus_2} \preceq \mathcal{F}_{\boxplus_1}$ ) iff:*

$$\bigcup \mathcal{F}_{\square_1} \supseteq \bigcup \mathcal{F}_{\square_2} \text{ and } \bigcup \mathcal{F}_{\blacksquare_1} \subseteq \bigcup \mathcal{F}_{\blacksquare_2} \text{ and } \forall_{B_2 \in \mathcal{F}_{\square_2}} \exists_{B_1 \in \mathcal{F}_{\square_1}} B_2 \subseteq B_1$$

Two joint box covers,  $\mathcal{F}_{\boxplus_1}$  and  $\mathcal{F}_{\boxplus_2}$ , are comparable iff  $\mathcal{F}_{\boxplus_1} \preceq \mathcal{F}_{\boxplus_2}$  or  $\mathcal{F}_{\boxplus_2} \preceq \mathcal{F}_{\boxplus_1}$ . Since not all joint box covers are comparable, the previous relation is a partial order.

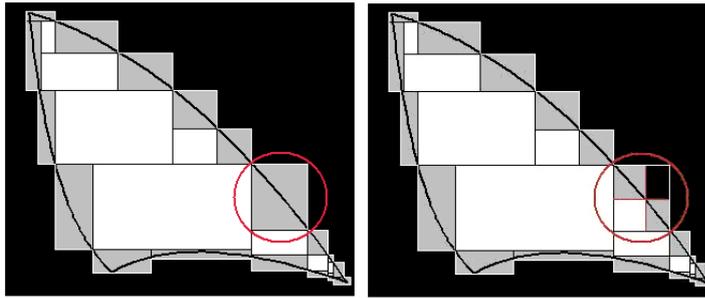
**Property 2.3 (Partial Order)** *Given a CCSP  $\langle X, D, C \rangle$ , the  $\preceq$  relation over joint box covers of  $\mathcal{F}$  is a partial order, since we have:*

$$\mathcal{F}_{\boxplus} \preceq \mathcal{F}_{\boxplus} \quad (\text{reflexivity})$$

$$\text{if } \mathcal{F}_{\boxplus_1} \preceq \mathcal{F}_{\boxplus_2} \quad \text{and} \quad \mathcal{F}_{\boxplus_2} \preceq \mathcal{F}_{\boxplus_1} \quad \text{then} \quad \mathcal{F}_{\boxplus_1} = \mathcal{F}_{\boxplus_2} \quad (\text{antisymmetry})$$

$$\text{if } \mathcal{F}_{\boxplus_1} \preceq \mathcal{F}_{\boxplus_2} \quad \text{and} \quad \mathcal{F}_{\boxplus_2} \preceq \mathcal{F}_{\boxplus_3} \quad \text{then} \quad \mathcal{F}_{\boxplus_1} \preceq \mathcal{F}_{\boxplus_3} \quad (\text{transitivity}).$$

Figure 2.3 shows the inner (white rectangles) and outer (white plus grey rectangles) box covers of the feasible space represented by the area inside the *curve lines*. In figure 2.3(b) the boundary box, marked with a circle in figure 2.3(a), is replaced by two smaller boundary boxes, one inner box and one non-solution box, providing a tighter joint box cover.



(a) Joint box cover

(b) Tighter joint box cover

**Figure 2.3:** Joint box covers of the feasible space inside the *curve lines*.

Given a joint box cover of a CCSP, the constraint reasoning step in algorithm 2 provides a way of computing a new tighter joint box cover.

A number of functions are input parameters to this algorithm: a) the *split* function defines how to partition a box into two or more sub-boxes; b) the *inner* predicate verifies whether a box is an inner box of the set of CCSP constraints; c) the *eligible* predicate checks whether a box is eligible for further processing; and d) the *order* function specifies which box, from the outer box cover, is retrieved for such processing.

The algorithm removes a box from the outer box cover that verifies the *eligible* predicate and is selected by the *order* function (line 1), and splits it (line 3). The algorithm subsequently modifies the inner and outer box covers of the joint cover. If the retrieved

**Algorithm 2:**  $crStep(\mathcal{F}_{\boxplus}, C, split, inner, eligible, order)$ 

**Input:**  $\mathcal{F}_{\boxplus}$ : CCSP joint box cover;  $C$ : set of constraints;  $split$ : function;  
 $inner, eligible$ : predicate;  $order$ : criteria;  
**Output:**  $\mathcal{F}_{\boxplus_{out}}$ : CCSP joint box cover;

- 1  $B \leftarrow remove(\mathcal{F}_{\square}, eligible, order)$ ;
- 2 **if**  $B = \emptyset$  **then return**  $\mathcal{F}_{\boxplus}$ ;
- 3  $S \leftarrow split(B)$ ;
- 4 **if**  $B \in \mathcal{F}_{\blacksquare}$  **then**
- 5  $\mathcal{F}_{\blacksquare} \leftarrow \mathcal{F}_{\blacksquare} \setminus \{B\}$ ;
- 6  $L_{\blacksquare} \leftarrow S$ ;
- 7  $L_{\square} \leftarrow S$ ;
- 8 **else**
- 9  $L_{\square} \leftarrow \{CPA(B_i, C) : B_i \in S\}$ ;
- 10  $L_{\blacksquare} \leftarrow \{B_i \in L_{\square} : inner(B_i, C)\}$ ;
- 11 **end**
- 12 **return**  $\langle \mathcal{F}_{\square} \cup L_{\square}, \mathcal{F}_{\blacksquare} \cup L_{\blacksquare} \rangle$ ;

box is already in the inner box cover (line 4) then it is replaced by the boxes resulting from the split, which are also added to the outer box cover<sup>1</sup> (lines 5 – 7). Otherwise (line 8) the boxes resulting from the split, are pruned by the constraint propagation algorithm and added to the outer box cover (line 9). Those that are inner boxes are also added to the inner box cover (line 10). The result is the modified joint box cover (line 12).

**Property 2.4 (Constraint Reasoning Step Result)** *Given a CCSP  $\langle X, D, C \rangle$  and a joint box cover  $\mathcal{F}_{\boxplus}$  of  $\mathcal{F}$ , the result of applying the constraint reasoning step to  $\mathcal{F}_{\boxplus}$  results in a tighter joint box cover, i.e.,  $crStep(\mathcal{F}_{\boxplus}, C, \cdot) \preceq \mathcal{F}_{\boxplus}$ .*

**Proof.** Let us denote by  $\mathcal{F}_{\boxplus_0}$  the original joint box cover and by  $\mathcal{F}_{\boxplus_1}$  the joint box cover resulting from applying the constraint reasoning step. By definition 2.33, we need to prove that  $\mathcal{F}_0^+ \supseteq \mathcal{F}_1^+$  and  $\mathcal{F}_0^- \subseteq \mathcal{F}_1^-$  and  $\forall B_2 \in \mathcal{F}_{\square_2} \exists B_1 \in \mathcal{F}_{\square_1} B_2 \subseteq B_1$ .

If the retrieved box  $B$  is the empty box then the algorithm stops by returning the original joint box cover and the conditions above hold.

<sup>1</sup>In fact this is an abstraction of the real implementation procedure that simply keeps a flag in each box signaling if it is an inner box.

Otherwise, if the retrieved box  $B$  is an inner box then, in both the inner and outer box covers,  $B$  is replaced by the set of sub-boxes resulting from the split,  $S$ . More formally,  $\mathcal{F}_1^+ = \mathcal{F}_0^+ \setminus B \cup \bigcup S$  and  $\mathcal{F}_1^- = \mathcal{F}_0^- \setminus B \cup \bigcup S$ . Since  $\bigcup S = B$  then  $\mathcal{F}_1^+ = \mathcal{F}_0^+$  and  $\mathcal{F}_1^- = \mathcal{F}_0^-$  and  $\forall_{B_i \in S} B_i \subseteq B$  and the conditions above hold.

Otherwise, each box resulting from the split is processed by the  $\mathcal{CPA}$  algorithm. In the outer box cover  $B$  is replaced by the boxes that result from this processing,  $L_\square$ . More formally  $\mathcal{F}_1^+ = \mathcal{F}_0^+ \setminus B \cup \bigcup L_\square$ . Since  $\mathcal{CPA}$  is a contracting algorithm then  $B \supseteq \bigcup L_\square$ . So  $\mathcal{F}_0^+ \supseteq \mathcal{F}_1^+$  and the first condition above holds. The inner box cover is augmented with the boxes resulting from the  $\mathcal{CPA}$  algorithm that are identified as inner boxes (possibly none),  $L_\blacksquare$ . More formally  $\mathcal{F}_1^- = \mathcal{F}_0^- \cup \bigcup L_\blacksquare$ . So, since  $\mathcal{F}_0^- \subseteq \mathcal{F}_1^-$ , the second condition above holds. Finally, since  $B$  was replaced by the boxes in  $L_\square$  that resulted from splitting  $B$  and applying a contracting algorithm to each of them, then  $\forall_{B_i \in L_\square} B_i \subseteq B$  and the third condition above holds.  $\blacksquare$

The constraint reasoning algorithm, presented in algorithm 3, applies repeatedly the constraint reasoning step until a stopping criterion is reached.

**Algorithm 3:**  $cReasoning(\mathcal{F}_\boxplus, C, split, inner, eligible, order, stop)$

<p><b>Input:</b> <math>\mathcal{F}_\boxplus</math>: CCSP joint box cover; <math>C</math>: set of constraints; <math>inner, eligible, stop</math>: predicates; <math>split</math>: function; <math>order</math>: criteria;</p> <p><b>Output:</b> <math>\mathcal{F}_{\boxplus_{out}}</math>: CCSP joint box cover;</p> <pre> 1 <math>\mathcal{F}'_\boxplus \leftarrow \mathcal{F}_\boxplus</math>; 2 <b>while</b> (<math>\neg stop(\mathcal{F}_\boxplus)</math>) <b>do</b> 3   <math>\mathcal{F}_\boxplus \leftarrow crStep(\mathcal{F}'_\boxplus, C, split, inner, eligible, order)</math>; 4   <b>if</b> <math>\mathcal{F}_\boxplus \neq \mathcal{F}'_\boxplus</math> <b>then</b> <math>\mathcal{F}'_\boxplus \leftarrow \mathcal{F}_\boxplus</math>; 5   <b>else</b> <b>break</b>; 6 <b>end</b> 7 <b>return</b> <math>\mathcal{F}_\boxplus</math>;                 </pre>
--

Algorithm 3 reasons over a given CCSP joint box cover by consecutively applying the  $crStep$  algorithm (lines 1 and 3 – 4) until the stopping criterion is reached (line 2) or there are no more eligible boxes to process in the outer box cover (lines 4 – 5). Then it returns the resulting joint box cover (line 7). The stopping criterion is imposed by the  $stop$  predicate given as input to this algorithm.

Notice that  $cReasoning$  (and ultimately  $crStep$ ) is parameterizable by using distinct  $inner, eligible$  and  $stop$  predicates,  $split$  functions and  $order$  criteria. Section A.1 of

appendix A, presents alternatives for parameterizing algorithms 2 and 3, resulting in distinct *cReasoning* versions used in this thesis. It also defines the default parametrization for the *inner* and *eligible* predicates and the *split* function.

To study the convergence of the *cReasoning* algorithm we assume that the algorithm is implemented with an infinite precision interval arithmetic.

Let  $\mathcal{F}_{\boxplus_k} = \langle \mathcal{F}_{\square_k}, \mathcal{F}_{\blacksquare_k} \rangle$  be the joint box cover computed at iteration  $k$  of the while loop in *cReasoning* and  $\Delta_k \mathcal{F} = \mathcal{F}_k^+ \setminus \mathcal{F}_k^-$ .

**Property 2.5 (Convergence)** *Let  $\langle X, D, C \rangle$  be a CCSP and  $cReasoning_\infty$  a family of *cReasoning* algorithms where the stop predicate returns false, the inner predicate is  $inner_d$ , the split function is fair and the conjunction of the order criterion and the eligible predicate imposes a fair choice wrt boundary boxes.*

*Consider a sequence  $(\mathcal{F}_{\boxplus_k})_{k \in \mathbb{N}}$  computed by  $cReasoning_\infty$  such that  $\mathcal{F}_{\boxplus_k} = crStep(\mathcal{F}_{\boxplus_{k-1}}, C, split, inner_d, true, order)$  and where  $\mathcal{F}_{\boxplus_0} = \langle \{D\}, \emptyset \rangle$  is the input joint box cover of  $cReasoning_\infty$ . Then*

$$\lim_{k \rightarrow \infty} vol(\Delta_k \mathcal{F}) = 0$$

**Proof.** See section A.2 in appendix A. ■

In practice, limitations of interval arithmetic precision imposes bounds to this theoretical convergence result.

To conclude, we illustrate the use of *cReasoning* algorithm with specific parametrization, to solve different problems (see appendix A for the default parametrization).

**Example 2.6.** Consider a CCSP  $\langle X, D, C \rangle$  for which we want to find one inner box. The pseudo-code for this function is given in algorithm 4. The *cReasoning* algorithm is used, with the  $order_{LIFO}$  criterion that induces the behavior of a *LIFO* data structure to  $\mathcal{F}_{\square}$ , and the *stop* predicate causes *cReasoning* to stop when the first inner box is found (line 1). All other arguments of *cReasoning* are parameterized by their defaults.

---

**Algorithm 4:**  $findInnerSolution(\langle X, D, C \rangle, \varepsilon)$ 

**Input:**  $\langle X, D, C \rangle$ : CCSP;  $\varepsilon$ : double;  
**Output:**  $B$ : box;  
**1**  $stop(\langle \mathcal{F}_\square, \mathcal{F}_\blacksquare \rangle) \equiv \mathcal{F}_\blacksquare \neq \emptyset$ ;  
**2**  $\langle \mathcal{F}_\square, \mathcal{F}_\blacksquare \rangle \leftarrow cReasoning(\langle \{D\}, \emptyset \rangle, C, split_2, inner_d, eligible_\varepsilon, order_{LIFO}, stop)$ ;  
**3** **return**  $getFirst(\mathcal{F}_\blacksquare)$ ;

**Example 2.7.** Consider the feasible space  $\mathcal{F}$  of a CCSP  $\langle X, D, C \rangle$ . Given a joint box cover  $\mathcal{F}_\boxplus = \langle \mathcal{F}_\square, \mathcal{F}_\blacksquare \rangle$  of  $\mathcal{F}$ , for any box  $B_i \in \mathcal{F}_\square$ , an enclosure for the volume of  $B_i \cap \mathcal{F}$  is:

$$vol(B_i \cap \mathcal{F}) \in [vol_{\mathcal{F}}](B_i) = \begin{cases} [vol(B_i)] & \text{if } B_i \in \mathcal{F}_\blacksquare \\ [0, vol(B_i)] & \text{otherwise} \end{cases}$$

and an enclosure for the volume of the feasible space is given by:

$$vol(\mathcal{F}) \in [vol](\mathcal{F}_\boxplus) = \sum_{B_i \in \mathcal{F}_\square} [vol_{\mathcal{F}}](B_i)$$

Algorithm 5 computes an enclosure for the volume of  $\mathcal{F}$  using  $cReasoning$ , where the  $stop_\delta$  predicate imposes a specified accuracy  $\delta$  for the volume enclosure computed over its argument (line 1); and the  $order_V$  criteria chooses boxes by decreasing order of the width of their volume enclosures. In fact, given the uncertainty on their volume enclosures, only boundary boxes are chosen for processing. The other arguments are parameterized by their defaults.

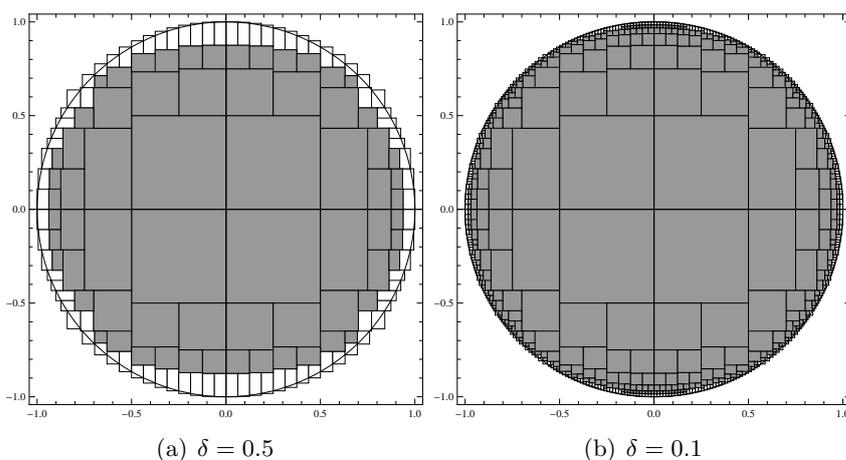
**Algorithm 5:**  $feasibleSpaceVolume(\langle X, D, C \rangle, \varepsilon, \delta)$ 

**Input:**  $\langle X, D, C \rangle$ : CCSP;  $\varepsilon, \delta$ : double  
**Output:**  $I$ : interval;  
**1**  $stop_\delta(\mathcal{F}_\boxplus) \equiv wid([vol](\mathcal{F}_\boxplus)) \leq \delta$ ;  
**2**  $\mathcal{F}_\boxplus \leftarrow cReasoning(\langle \{D\}, \emptyset \rangle, split_2, inner_d, eligible_\varepsilon, order_V, stop_\delta)$ ;  
**3** **return**  $[vol](\mathcal{F}_\boxplus)$ ;

Now consider the CCSP *Circle* where  $X = \langle x, y \rangle$ ,  $D = [-1, 1] \times [-1, 1]$  and  $C = \{C_1\}$  with  $C_1 : x^2 + y^2 \leq 1$ . The exact volume of the *Circle* feasible space is  $\pi$ .

Figures 2.4 (a) and (b) show the joint box covers resulting from applying algorithm 5 to *Circle*,  $\varepsilon = 10^{-8}$  and, respectively,  $\delta = 0.5$  and  $\delta = 0.1$ . The enclosures obtained for the volume were  $[2.720664277013146, 3.21952974686935]$  for  $\delta = 0.5$  with a total

of 152 boxes and  $[3.060639819475821, 3.160525625764372]$  for  $\delta = 0.1$  with a total of 744 boxes. An extra test was made for  $\delta = 0.0001$  (not shown in the figure) and the obtained enclosure for the volume was  $[3.141511312676047, 3.141611312486993]$  with a total of 725265 boxes.



**Figure 2.4:** Joint box covers obtained when computing the volume of a circle with radius 1, centered in  $(0,0)$ .

## 2.4 Summary

This chapter focused on the core concepts of continuous constraint programming, by introducing interval analysis (interval arithmetic, inclusion functions and interval methods), continuous constraint satisfaction problems and methods to solve them (constraint propagation and constraint reasoning). Although describing classical notions and techniques, the topics are biased towards the probabilistic continuous constraint framework proposed in the thesis.

In the next chapter relevant concepts of probability theory, the other component of the framework, will be introduced.



## CHAPTER 3

# Probabilistic Uncertainty Quantification

Although many other general models of uncertainty have been developed (e.g. imprecise probabilities [116], Dempster-Shafer theory of evidence [109] and random sets [72], fuzzy sets [120], possibility theory [2], probability bounds [45], convex model [12], and others) probability theory is the traditional approach to handle uncertainty that we have adopted in this thesis.

The thesis combines two models of uncertainty: a) ranges of possible values for uncertain variables, and b) probability distributions over those ranges. This chapter overviews the foundations of probability theory by describing its central objects: events and random variables, which are mathematical abstractions of non-deterministic events or measured quantities. In particular, we are interested in events occurring in continuous sample spaces and in continuous random variables. In section 3.1 the theoretical concepts of probability theory are introduced, followed by section 3.2, which presents the concept of conditional probability. Next, sections 3.3 and 3.4 describe, respectively, random variables and random vectors as well as their properties. Finally, section 3.5 presents two techniques to compute integrals over boxes: the first based on Taylor models, results in a guaranteed enclosure for the integral value and the other, based on Monte Carlo integration, results in an approximate enclosure for that value.

## 3.1 Probability

Probability theory studies models of random phenomena intended to describe random experiments, i.e., experiments that can be repeated (indefinitely) and where future

outcomes cannot be exactly predicted even if the experimental situation can be fully controlled, given the underlying randomness affecting such experiments.

**Definition 3.1 (Sample Space)** *The sample space  $\Omega$  is the set of all possible distinct outcomes of a random experiment.*

Sample spaces may be finite, countable infinite, or uncountable infinite. Intuitively, an event is the outcome or set of outcomes of a random experiment that share a common attribute.

**Definition 3.2 (Event)** *An event  $H$  in a sample space  $\Omega$  is a measurable subset of  $\Omega$ , i.e.,  $H \subseteq \Omega$ .*

Each outcome,  $\omega$ , of an experiment is represented by a point in  $\Omega$  and is called a sample point. An event on the sample space is represented by an appropriate collection of sample points. The event containing a single sample point is called atomic event.

When the sample space is finite (or even countable infinite) probabilities on every possible subset of this space can be defined. However, this is not so for uncountably infinite sample spaces.

The Vitali set is an example of a subset of  $\mathbb{R}$  for which it is not possible to define a probability measure. This set is so complicated that if we try to define a probability measure on it, we will run into contradiction [101, p. 22] and [103, p. 698 - 671]. This is why we need to restrict the definition of events to certain collection of subsets, denoted as  $\sigma$ -algebras.

**Definition 3.3 ( $\sigma$ -Algebra)** *A  $\sigma$ -algebra  $\mathcal{A}$  on a set  $\Omega$  is a non-empty collection of subsets of  $\Omega$ , that is closed under complements and countable unions of its members, as defined below.*

$$\begin{aligned} \mathcal{A} &\subseteq 2^\Omega \\ H \in \mathcal{A} &\Rightarrow \bar{H} \in \mathcal{A}, \quad \text{where } \bar{H} \text{ denotes the complement of } H \\ \forall_{i \in I} H_i \in \mathcal{A} &\Rightarrow \bigcup_{i \in I} H_i \in \mathcal{A} \quad \text{where } I \text{ is a countable index set} \end{aligned}$$

**Property 3.1 ( $\sigma$ -Algebra Properties)** *From the previous axioms, a  $\sigma$ -algebra on  $\Omega$  contains  $\Omega$  itself, the empty set, and is closed under countable intersections of its members, i.e.,*

$$\emptyset \in \mathcal{A}$$

$$\Omega \in \mathcal{A}$$

$$\forall_{i \in I} H_i \in \mathcal{A} \Rightarrow \bigcap_{i \in I} H_i \in \mathcal{A} \quad \text{where } I \text{ is a countable index set}$$

We are now able to define a measurable space.

**Definition 3.4 (Measurable Space)** *A measurable space is a pair  $\langle \Omega, \mathcal{A} \rangle$ , where  $\mathcal{A}$  is a  $\sigma$ -algebra on  $\Omega$ .*

*In this context, a subset of  $\Omega$  is an event iff it belongs to  $\mathcal{A}$ .*

If  $\Omega$  is countable then  $\mathcal{A}$  can be defined as the power set of  $\Omega$ , i.e.  $\mathcal{A} = 2^\Omega$ , which is trivially a  $\sigma$ -algebra and the largest that can be created using  $\Omega$ . When dealing with uncountable sets we can end up with an *untractable* set if we use the power set. However, when  $\Omega \subseteq \mathbb{R}$  ( $\mathbb{R}$  is an uncountable infinite set) there is an adequate  $\sigma$ -algebra on  $\mathbb{R}$ , known as Borel  $\sigma$ -algebra.

**Definition 3.5 (Borel  $\sigma$ -Algebra)** *The Borel  $\sigma$ -algebra on  $\mathbb{R}$ , represented by  $\mathcal{B}$ , is generated by all open (or closed) intervals and is a standard (or implicit)  $\sigma$ -algebra on Euclidean spaces.*

When  $S_1, \dots, S_n$  are sets and  $\mathcal{A}_i$   $\sigma$ -algebras for each  $S_i$  then a  $\sigma$ -algebra  $\mathcal{A}^n$  can be used for product set  $S^n = S_1 \times \dots \times S_n$ , generated by the collection of all product sets of the form  $A_1 \times \dots \times A_n$  where  $A_i \in \mathcal{A}_i$  for each  $i$ . Hence, the notion of Borel  $\sigma$ -algebra can be generalized to  $\mathbb{R}^n$ , denoted as  $\mathcal{B}^n$ .

We now define a probability measure with respect to the sample space  $\Omega$ .

**Definition 3.6 (Probability Measure)** *Given a measurable space  $\langle \Omega, \mathcal{A} \rangle$ , the probability measure (or simply, probability)  $P(\cdot)$  is a function that maps sets in  $\mathcal{A}$  into the set of real numbers and satisfies the following three axioms:*

$$\mathbf{A1} : \forall H \in \mathcal{A} \quad P(H) \geq 0$$

$$\mathbf{A2} : P(\Omega) = 1$$

$$\mathbf{A3} : P\left(\bigcup_{i=1}^{\infty} H_i\right) = \sum_{i=1}^{\infty} P(H_i) \quad \text{where } \forall_i H_i \in \mathcal{A} \text{ and } \forall_{i \neq j} H_i \cap H_j = \emptyset$$

The third axiom is known as countable additivity, and states that the probability of a union of a finite or countably infinite collection of disjoint events is the sum of the corresponding probabilities.

The concepts of sample spaces,  $\sigma$ -algebras and probability measure can be combined in the definition of probability space.

**Definition 3.7 (Probability Space)** *A probability space is a triple  $\langle \Omega, \mathcal{A}, P \rangle$  where  $\Omega$  is the sample space,  $\mathcal{A}$  is a  $\sigma$ -algebra on  $\Omega$ , and  $P$  is the probability measure defined on the measurable space  $\langle \Omega, \mathcal{A} \rangle$ .*

The definition of probability measure implies some properties of the corresponding function  $P(\cdot)$  as shown below.

**Property 3.2 (Probability Measure Properties)** *Any probability measure  $P(\cdot)$  has the following properties for any two events  $A$  and  $B$ :*

- $P(\emptyset) = 0$
- $P(A) = 1 - P(\bar{A})$
- If  $A \subseteq B$  then  $P(A) \leq P(B)$
- $P(A) \in [0, 1]$     *Numeric bounds of probability*
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$     *General additivity*

- $P(\bigcup_{i=1}^n B_i) \leq \sum_{i=1}^n P(B_i)$  *Boole's inequality*

Consider  $n$  events  $A_1, \dots, A_n$ , then the inclusion-exclusion principle states

$$P\left(\bigcup_{i=1}^n A_i\right) = \sum_{i=1}^n P(A_i) - \sum_{i,j:i<j} P(A_i \cap A_j) \\ + \sum_{i,j,k:i<j<k} P(A_i \cap A_j \cap A_k) - \dots + (-1)^{n-1} P\left(\bigcap_{i=1}^n A_i\right)$$

The general additivity and Boole's inequality properties are derived from the inclusion-exclusion principle.

The countable additivity property, which assumes disjoint events, is adapted to almost disjoint events, as follows.

**Definition 3.8 (Almost Disjoint Events)** Two events  $H_1$  and  $H_2$  in  $\mathcal{B}^n$  are almost disjoint iff  $P(H_1 \cap H_2) = 0$ .

**Property 3.3 (Probability of an Almost Disjoint Set of Events)** Given a mutually almost disjoint set of events<sup>1</sup>  $\mathcal{S} = \{S_1, \dots, S_n\}$  such that  $\forall_i S_i \in \mathcal{B}^n$ :

$$P\left(\bigcup_{i=1}^n S_i\right) = \sum_{i=1}^n P(S_i)$$

**Proof.** Directly from the inclusion-exclusion principle (in property 3.2) and the definition of almost disjoint events. ■

## 3.2 Conditional Probability

Probabilistic reasoning [102] aims at incorporating new information, known as evidence, by updating an *a priori* probability describing what is known in the absence of the

<sup>1</sup>A set of events is mutually almost disjoint iff, given any two events in the set, they are almost disjoint.

evidence into an *a posteriori* probability given the evidence. For incorporating this evidence, conditioning is used. The probability of some event  $A$ , given the occurrence of some other event  $B$  is denoted as  $P(A|B)$ . The *a posteriori* probability is the conditional probability when the evidence is taken into account.

More formally, let  $\Omega$  be a sample space and let  $P(\cdot)$  denote the probability assigned to the events in  $\Omega$ . These probabilities should be revised once an event  $B$  has occurred into the conditional probability,  $P(\cdot|B)$ . The conditional probability measure must satisfy the fundamental properties, stated in definition 3.6, required to any probability measure. Moreover, since the probability of a sure event must be 1, then the probability of  $B$  given  $B$  must be 1. Similarly, since the probability of an impossible event is zero, the probability of any event disjoint from  $B$  must be zero.

Given two events,  $A_1 \subseteq B$  and  $A_2 \subseteq B$ , if  $A_1$  is  $k$  times more likely than  $A_2$  before receiving the information  $B$ , then it remains  $k$  times more likely after receiving such information. This is because all the outcomes in  $A_1$  and  $A_2$  remain possible and, hence, there is no reason to expect that the ratio of their likelihoods changes.

The previous properties are summarized, in a more formal way, as follows.

**Property 3.4 (Conditional Probability Measure Properties)** *Given a probability space  $\langle \Omega, \mathcal{A}, P \rangle$  and evidence  $B \in \mathcal{A}$ , with  $P(B) > 0$ :*

1. **Probability measure.**  $P(\cdot|B)$  satisfies all the properties of a probability measure.

$$(a) \quad \forall A \in \mathcal{A} P(A|B) \geq 0$$

$$(b) \quad P(\Omega|B) = 1$$

$$(c) \quad P\left(\bigcup_{i=1}^{\infty} A_i|B\right) = \sum_{i=1}^{\infty} P(A_i|B) \quad \text{where } \forall_i A_i \in \mathcal{A} \text{ and } \forall_{i \neq j} A_i \cap A_j = \emptyset$$

2. **Sure event.**  $P(B|B) = 1$ .

3. **Impossible events.** If  $A \subseteq \overline{B}$  then  $P(A|B) = 0$ .

4. **Constant likelihood ratios.** If  $A_1 \subseteq B$  and  $A_2 \subseteq B$  (with  $P(A_2) > 0$ ), then

$$\frac{P(A_1)}{P(A_2)} = \frac{P(A_1|B)}{P(A_2|B)}.$$

The definition of conditional probability  $P(\cdot|B)$  can now be derived, from the absolute probability  $P(\cdot)$  defined in  $\Omega$  (note that  $P(\cdot) = P(\cdot|\Omega)$ ), for any given event  $A \subseteq \mathcal{A}$ .

From property (4) above, and considering  $A_1 = A$  and  $A_2 = B$ , follows that  $\frac{P(A)}{P(B)} = \frac{P(A|B)}{P(B|B)}$ . From property (2), this can be rewritten as  $\frac{P(A)}{P(B)} = P(A|B)$ , for any  $A \subseteq B$ .

Now consider an event  $A$  not totally contained in  $B$ .

$$\begin{aligned}
 A &= (A \cap B) \cup (A \cap \bar{B}) && \text{set algebra.} \\
 P(A|B) &= P(A \cap B|B) + P(A \cap \bar{B}|B) && (A \cap B) \cap (A \cap \bar{B}) = \emptyset. \\
 P(A \cap \bar{B}|B) &= 0 && (A \cap \bar{B}) \subseteq \bar{B} \text{ and } P(\bar{B}|B) = 0. \\
 P(A|B) &= \frac{P(A \cap B)}{P(B)} && (A \cap B) \subseteq B.
 \end{aligned}$$

Hence, the definition.

**Definition 3.9 (Conditional Probability)** *Given a probability space  $(\Omega, \mathcal{A}, P)$ , if  $A, B \in \mathcal{A}$  are two events with  $P(B) > 0$  the conditional probability of  $A$  given  $B$  is defined as:*

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

We can then conclude that, for any event  $A \in \mathcal{A}$  and evidence  $B \in \mathcal{A}$ , the conditional probability measure  $P(A \cap B|B)(= P(A|B))$  is related to the probability measure  $P(A \cap B)$  by a constant scale factor  $\alpha = \frac{1}{P(B)}$ .

Given two events  $A$  and  $B$  in some situations  $P(A|B) = P(A)$ , i.e., the knowledge of  $B$  does not affect the probability of  $A$ . In this case we say that  $A$  and  $B$  are independent and hence the appropriate definition of independent events is  $P(A \cap B) = P(A)P(B)$ , which can be generalized as follows.

**Definition 3.10 (Mutually Independent Events)** *The events of a set  $\{A_1, \dots, A_n\}$  are mutually independent iff all subsets  $\{A_{\nu_1}, \dots, A_{\nu_k}\}$  of these*

events verify

$$P\left(\bigcap_{i=1}^k A_{\nu_i}\right) = \prod_{i=1}^k P(A_{\nu_i})$$

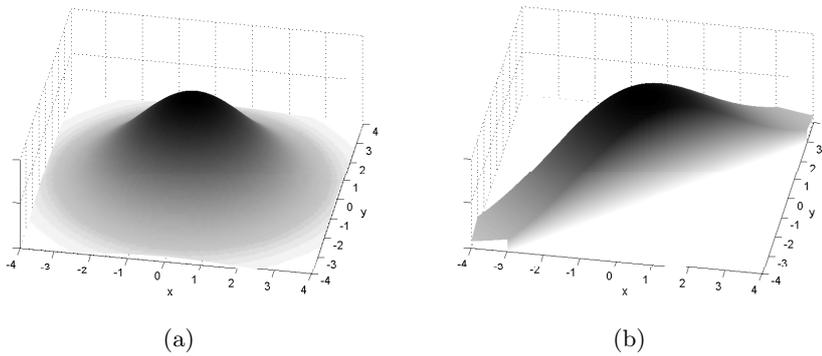
Given a probability space  $\langle \Omega, \mathcal{A}, P \rangle$  and evidence  $B$ , a new probability space can be defined as follows.

**Definition 3.11 (Conditional Probability Space)** *Given a probability space  $(\Omega, \mathcal{A}, P)$ , let  $B \in \mathcal{A}$  be some event with  $P(B) > 0$ . On the same measurable space  $\langle \Omega, \mathcal{A} \rangle$ , let the probability measure  $Q$  be defined as:*

$$Q(A) = P(A|B)$$

*Then  $(\Omega, \mathcal{A}, Q)$  is also a probability space and is called a conditional probability space.*

**Example 3.1.** Consider a probability space where  $\Omega = \mathbb{R}^2$  characterized by a bivariate Gaussian distribution. Consider as *evidence* the event defined by the set of points that satisfy the inequalities  $x - 1 \leq y \leq x + 1$ . Figure 3.1 (a) illustrates the probability space distribution when the evidence is not considered and figure 3.1 (b) the conditional probability space distribution given the evidence.

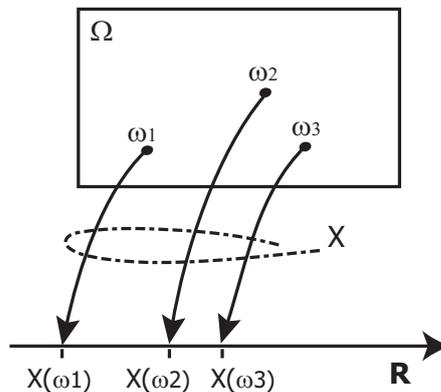


**Figure 3.1:** (a) Bivariate Gaussian distributed probability space; (b) Conditional probability space distribution given the evidence  $x - 1 \leq y \leq x + 1$ .

### 3.3 Random Variables

Intuitively, a random variable  $X$  is a measurement of interest in some random experiment. A variable  $X$  is random in the sense that its value depends on the outcome of the experiment, which cannot be predicted with certainty before the experiment is run. Each time the experiment is run, an outcome  $\omega \in \Omega$  occurs, and a given random variable  $X$  takes on the value  $X(\omega)$ .

A real-valued random variable is a function from the sample space  $\Omega$  into the real numbers, i.e., it assigns a real number to every element of the sample space, as shown in figure 3.2.



**Figure 3.2:** Mapping from the sample space  $\Omega$  to  $\mathbb{R}$ , by the random variable  $X$ .

**Definition 3.12 (Random Variable)** A function  $X : \Omega \rightarrow \mathbb{R}$  is a real-valued random variable defined on the probability space  $\langle \Omega, \mathcal{A}, P \rangle$  iff, for any  $S \in \mathcal{B}$ , we have:

$$\{\omega \in \Omega : X(\omega) \in S\} \in \mathcal{A}$$

Note that a statement about a random variable defines an event. We adopt the notation  $\{X \in S\}$  to express, more compactly, the event induced via the random variable  $X$ .

We can notice that, if the outcome of an experiment is in  $\mathbb{R}^n$ , the projection to any of its coordinates is also a random variable.

**Definition 3.13 (Projection Random Variable)** *A random variable  $X_i : \Omega \rightarrow \mathbb{R}$ , with  $\Omega \subseteq \mathbb{R}^n$ , is a projection random variable wrt coordinate  $i$  if  $X_i(\omega_1, \dots, \omega_n) = \omega_i$ .*

**Property 3.5 (Function of a Random Variable)** *Let  $X$  be a random variable defined on the probability space  $\langle \Omega, \mathcal{A}, P \rangle$  and  $g : \mathbb{R} \rightarrow \mathbb{R}$  a Borel measurable function<sup>1</sup>. Then  $Y = g(X)$  is also a random variable.*

To specify probability measures for random variables, it is often convenient to adopt functions from which the probability measure immediately follows, namely the cumulative distribution function (CDF) and the probability density function (PDF).

The cumulative distribution function describes the probability that a random variable  $X$ , with a given probability distribution, takes a value less or equal than any  $x \in \mathbb{R}$ .

**Definition 3.14 (Cumulative Distribution Function)** *The CDF of a random variable  $X$ , is a function  $F_X : \mathbb{R} \rightarrow [0, 1]$ , which specifies a probability measure as:*

$$F_X(x) = P(\{X \leq x\})$$

**Property 3.6 (CDF Properties)** *For any  $x, y \in \mathbb{R}$ :*

$$\begin{aligned} 0 &\leq F_X(x) \leq 1 \\ \lim_{x \rightarrow -\infty} F_X(x) &= 0 \\ \lim_{x \rightarrow +\infty} F_X(x) &= 1 \\ x \leq y &\Rightarrow F_X(x) \leq F_X(y) \end{aligned}$$

A random variable is a continuous random variable (the main focus of this thesis) if its CDF  $F_X(\cdot)$  is absolutely continuous. Hence a probability density function (PDF) may be defined.

---

<sup>1</sup>A real-valued function  $g$  such that the inverse image of the set of real numbers greater than any given real number  $x$ , is a Borel set, i.e.  $g^{-1}([x, +\infty]) \in \mathcal{B}$ .

**Definition 3.15 (Probability Density Function)** For a continuous random variable  $X$ , with CDF,  $F_X(\cdot)$ , differentiable everywhere, its PDF is given by:

$$f_X(x) = \frac{dF_X(x)}{dx}$$

From the previous definition, we can conversely retrieve the CDF of a random variable  $X$ , from its PDF

$$F_X(x) = \int_{-\infty}^x f_X(u) du$$

According to the properties of differentiation, for very small  $\Delta x$ ,

$$P(\{X \in [x, x + \Delta x]\}) = \int_x^{x+\Delta x} f_X(x) dx \approx f_X(x) \Delta x$$

Thus,  $f_X(x)$  is not the probability of that event, i.e.  $f_X(x) \neq P(\{X = x\})$ , but rather the probability mass per unit area near  $x$ , representing a likelihood in its neighborhood.

The following properties can be derived from the CDF properties in property 3.6.

**Property 3.7 (PDF Properties)** A PDF  $f_X(\cdot)$  has the following properties:

$$\begin{aligned} f_X(x) &\geq 0 \\ \int_{-\infty}^{\infty} f_X(x) dx &= 1 \end{aligned}$$

The next statement defines the probability of an event induced via a random variable.

**Definition 3.16 (Probability of an Event)** Given a random variable  $X$  on the probability space  $(\Omega, \mathcal{A}, P)$  with PDF  $f_X(\cdot)$  and a set  $S \in \mathcal{B}$ , the probability of the event  $H = \{\omega \in \Omega : X(\omega) \in S\}$  is given by:

$$\int_S f_X(x) dx = P(\{X \in S\}) = P(H)$$

In particular, if  $S = [a, b]$ , the probability of event  $H = \{\omega \in \Omega : a \leq X(\omega) \leq b\}$ , is  $P(H) = \int_a^b f_X(x)dx$ .

Since the number of possible outcomes of a continuous random variable is uncountable infinite, the probability that  $X$  takes any single value  $a$  (that is  $a \leq X \leq a$ ) is zero<sup>1</sup> and probabilities for a continuous random variable over intervals should be considered. This probability can be interpreted as the area under the graph of the PDF between the interval bounds.

Although the probability that a continuous random variable takes a specific value is zero, this does not necessarily mean that such value cannot occur. In fact it means that the point (event) is one of an infinite number of possible outcomes. Formally, each value has an infinitesimally small probability, which statistically is equivalent to zero [105, p. 64].

### 3.3.1 Moments

In many situations we do not need all the detail on the randomness patterns in-built in the PDF (or CDF). A rather limited number of raw moments (or central moments) supply enough information for our purposes. Namely, under very broad conditions, an approximation using a Gaussian distribution (central limit theorem) is efficient.

Any probability distribution has a set of numerical characteristics associated with it (such as the expected value, the variance, the skewness, etc.), related to the moments of the probability density function.

The first moment of the PDF (centered at zero) is the expected value, i.e., the population mean, denoted by  $E[X]$  or, more compactly, by  $\mu$ . It is a measure that corresponds to the physical concept of center of mass of the distribution.

**Definition 3.17 (Expected Value)** *Given a random variable  $X$  with PDF  $f_X(\cdot)$  the expected value of  $X$  is*

$$E[X] = \int_{\mathbb{R}} x f_X(x) dx$$

---

<sup>1</sup>It is assumed that the PDF  $f(\cdot)$  is bounded, since the CDF is differentiable everywhere.

provided that  $\int_{\mathbb{R}} |x|f_X(x)dx < \infty$ .

Sometimes we are interested in computing the expected value of a random variable  $Y = g(X)$  when the probability distribution of  $X$  is known and the probability distribution of  $Y$  is not explicitly known.

**Definition 3.18 (Law of the Unconscious Statistician)** *Given a random variable  $X$  on the probability space with PDF  $f_X(\cdot)$  and a random variable  $Y = g(X)$  then, the expected value of  $Y$  is*

$$E[Y] = \int_{\mathbb{R}} g(x)f_X(x)dx$$

provided that  $\int_{\mathbb{R}} |g(x)|f_X(x)dx < \infty$ .

Notice, however, that given the random variables  $X$  and  $Y = g(X)$  the CDF (and PDF) of  $Y$  may be obtained using standard techniques [105, sec. 2.5].

**Property 3.8 (Expected Value Properties)** *Given random variable  $X$  and  $Y$  on the same probability space, the expected value has the following properties:*

$$E[a] = a \quad \text{for any constant } a \in \mathbb{R}$$

$$E[aX] = aE[X] \quad \text{for any constant } a \in \mathbb{R}$$

$$E[X + Y] = E[X] + E[Y] \quad (\text{Linearity of the Expected Value})$$

The second moment of the PDF (centered at its mean  $\mu$ ), is the variance, denoted by  $Var[X]$  or, more compactly, by  $\sigma^2$ . The variance and the closely-related standard deviation are measures of how concentrated the distribution is around its mean.

**Definition 3.19 (Variance and Standard Deviation)** *Given a random variable  $X$  on the probability space  $\langle \Omega, \mathcal{A}, P \rangle$  with PDF  $f_X(\cdot)$ , the variance of  $X$  is*

defined as:

$$\text{Var}[X] = E[(X - E[X])^2]$$

The standard deviation is given by  $\sigma = \sqrt{\text{Var}[X]}$ .

From the properties of the expected value an alternative expression for the variance follows.

**Property 3.9 (Computational Expression for the Variance)** *From the properties of the expected value we have that:*

$$\text{Var}[X] = E[X^2] - E[X]^2$$

### 3.3.2 Some Continuous Probability Distributions

Many probability distributions are so important in theory or applications that they have been given specific names. In the following we describe the main continuous probability distributions used in this thesis.

The uniform distribution assigns equal probability density to every value in a given range. A random variable  $X$  uniformly distributed in the range  $[a, b]$  is denoted by  $X \sim \mathcal{U}(a, b)$  and has PDF:

$$f_X(x) = \frac{1}{b-a} \quad x \in [a, b]$$

Possibly the most common distribution, used as a first approximation to describe random variables that tend to cluster around a single mean value, is the normal (or Gaussian) distribution. A random variable  $X$  normally distributed is denoted by  $X \sim \mathcal{N}(\mu, \sigma^2)$ , where  $\mu$  is the mean and  $\sigma^2$  the variance and has PDF:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}, \quad x \in \mathbb{R}$$

The graph of this function is *bell-shaped*. When  $\mu = 0$  and  $\sigma^2 = 1$  the distribution is called standard normal.

A random variable  $X$  distributed accordingly to a cosine distribution in the range  $[\mu - a, \mu + a]$  is denoted by  $X \sim \mathcal{C}(\mu, a)$ , where  $\mu$  is the mean point and  $a$  defines the range of the distribution and has PDF:

$$f_X(x) = \frac{1}{2a} \left[ 1 + \cos \left( \frac{x - \mu}{a} \pi \right) \right], \quad x \in [\mu - a, \mu + a]$$

A cosine distribution  $\mathcal{C}(\mu, a)$  is related to a triangular distribution,  $\mathcal{T}(\mu, \mu - a, \mu + a)$ , but its PDF is differentiable everywhere.

### 3.4 Random Vectors

In many situations we may be interested in dealing with several random variables.

**Definition 3.20 (Random Vector)** A vector-valued function  $\mathbf{X} : \Omega \rightarrow \mathbb{R}^n$  is a random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  defined on the probability space  $(\Omega, \mathcal{A}, P)$ , iff, for any  $S = S_1 \times \dots \times S_n \in \mathcal{B}^n$  we have:

$$\{\omega \in \Omega : X_1(\omega) \in S_1 \wedge \dots \wedge X_n(\omega) \in S_n\} \in \mathcal{A}$$

Each component of the vector is a real-valued random variable on the same probability space  $(\Omega, \mathcal{A}, P)$ .

The event induced via the random vector will be expressed, compactly, by  $\{\mathbf{X} \in S\}$ .

**Property 3.10 (Function of a Random Vector)** Let  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  be a random vector defined on the probability space  $(\Omega, \mathcal{A}, P)$  and  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$  a Borel measurable function<sup>1</sup>. Then  $\mathbf{Y} = g(\mathbf{X})$  is an  $m$ -dimensional random vector.

The realization of the random vector is denoted by  $\mathbf{x} = \langle x_1, \dots, x_n \rangle$ . Nevertheless, when used explicitly as the argument of a function, the brackets will be omitted.

<sup>1</sup>A function such that the inverse image of any  $m$ -dimensional Borel set is an  $n$ -dimensional Borel set, i.e., for any  $B \in \mathcal{B}^m$  we have that  $g^{-1}(B) \in \mathcal{B}^n$ .

When  $\Omega$  is a subset of  $\mathbb{R}^n$ , it is useful to consider the identity random vector that represents the outcome of the experiment itself.

**Definition 3.21 (Identity Random Vector)** *A random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  is an identity random vector if each  $X_i$  is its real-valued projection random variable wrt coordinate  $i$ .*

*Specifically, points of  $\mathbb{R}^n$  have the form  $\langle \omega_1, \dots, \omega_n \rangle$ , where  $\omega_i \in \mathbb{R}$ , so  $\mathbf{X}(\omega_1, \dots, \omega_n) = \langle X_1(\omega_1, \dots, \omega_n), \dots, X_n(\omega_1, \dots, \omega_n) \rangle = \langle \omega_1, \dots, \omega_n \rangle$ .*

The concepts presented in the previous section can be extended to random vectors as we will show next.

**Definition 3.22 (Joint Cumulative Distribution Function)** *Given a random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$ , its joint CDF is defined by:*

$$F_{\mathbf{X}}(x_1, \dots, x_n) = P(\{X_1 \leq x_1 \wedge \dots \wedge X_n \leq x_n\})$$

*A more compact way to express this is  $F_{\mathbf{X}}(\mathbf{x}) = P(\{\mathbf{X} \leq \mathbf{x}\})$ ,  $\mathbf{x} \in \mathbb{R}^n$ .*

From the joint CDF of a random vector the marginal cumulative distribution functions can be defined.

**Definition 3.23 (Marginal Cumulative Distribution Function)** *Given a random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  with joint CDF  $F_{\mathbf{X}}(\cdot)$ , the marginal CDF of each  $X_i$ ,  $F_{X_i}(\cdot)$ , is defined as:*

$$F_{X_i}(x_i) = F_{\mathbf{X}}(\infty, \dots, \infty, x_i, \infty, \dots, \infty)$$

Similarly, a marginal cumulative distribution can be derived from the random vector for any subset of its random variables (allowing the other variables to take any values).

The concept of probability density function can also be extended to random vectors.

**Definition 3.24 (Joint Probability Density Function)** Given a random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  with CDF  $F_{\mathbf{X}}(\cdot)$ , everywhere differentiable in every  $x_i$ , the joint PDF of  $\mathbf{X}$  is defined by:

$$f_{\mathbf{X}}(x_1, \dots, x_n) = \frac{\partial^n F_{\mathbf{X}}(x_1, \dots, x_n)}{\partial x_1 \dots \partial x_n}$$

From this definition, the joint CDF of a random vector, can be retrieved from its joint PDF by integration:

$$F_{\mathbf{X}}(x_1, \dots, x_n) = \int_{-\infty}^{x_n} \dots \int_{-\infty}^{x_1} f_{\mathbf{X}}(u_1, \dots, u_n) du_1 \dots du_n$$

For simplicity, when  $S = S_1 \times \dots \times S_n \subseteq \mathbb{R}^n$ , the symbol  $\int_S$  will be used to represent  $n$  consecutive integrations. The  $i^{\text{th}}$  integration region is defined by  $S_i$ , with  $1 \leq i \leq n$ . When  $S = \mathbb{R}^n$  the integrations are from  $-\infty$  to  $\infty$ .

Like in the single dimensional case,  $f_{\mathbf{X}}(x_1, \dots, x_n) \neq P(X_1 = x_1, \dots, X_n = x_n)$ . Nevertheless we can use the joint PDF of a random vector to compute the probability of an event.

**Definition 3.25 (Probability of an Event)** Given a random vector  $\mathbf{X}$  on the probability space  $(\Omega, \mathcal{A}, P)$  with PDF  $f_{\mathbf{X}}(\cdot)$  and a set  $S \in \mathcal{B}^n$ , the probability of the event  $H = \{\omega \in \Omega : \mathbf{X}(\omega) \in S\}$  is given by:

$$P(H) = P(\{\mathbf{X} \in S\}) = \int_S f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad \mathbf{x} \in \mathbb{R}^n$$

In particular, if  $S = [a_1, b_1] \times \dots \times [a_n, b_n]$ , the probability of the associated event  $H$  is given by  $P(H) = \int_{a_n}^{b_n} \dots \int_{a_1}^{b_1} f_{\mathbf{X}}(x_1, \dots, x_n) dx_1 \dots dx_n$ .

As for CDFs, from the joint PDF of the random vector marginal PDFs for each component random variable are obtained by *integrating out* the unwanted dimensions.

**Definition 3.26 (Marginal Probability Density Function)** *Given a random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  with PDF  $f_{\mathbf{X}}(\cdot)$ , then  $f_{X_i}(\cdot)$  is the marginal PDF of each  $X_i$ , defined as:*

$$f_{X_i}(x_i) = \int_{\mathbb{R}^{n-1}} f_{\mathbf{X}}(x_1, \dots, x_n) dx_1 \dots dx_{i-1} dx_{i+1} dx_n$$

Marginal joint PDFs can be extended to any combination of  $k$  component random variables, with  $k \leq n$ , obtaining the joint PDF for the  $k$  random variables.

In the special case where the random variables are independent their joint density may be computed from their individual densities.

**Definition 3.27 (Independent Random Variables)** *Let  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  be a random vector on the probability space  $(\Omega, \mathcal{A}, P)$  with marginal PDFs  $f_{X_1}(\cdot) \dots, f_{X_n}(\cdot)$ . The  $X_i$  are independent random variables iff their joint PDF is the product of their marginal PDFs:*

$$f_{\mathbf{X}}(x_1, \dots, x_n) = f_{X_1}(x_1) \dots f_{X_n}(x_n)$$

### 3.4.1 Moments

Expected values can be defined for random vectors, as follows.

**Definition 3.28 (Expected Value)** *The expected value of a random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  with joint PDF  $f_{\mathbf{X}}(\cdot)$  is:*

$$\begin{aligned} E[\mathbf{X}] &= \langle E[X_1], \dots, E[X_n] \rangle = \left\langle \int_{\mathbb{R}^n} x_1 f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}, \dots, \int_{\mathbb{R}^n} x_n f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \right\rangle \\ &= \left\langle \int_{\mathbb{R}} x_1 f_{X_1}(x_1) dx_1, \dots, \int_{\mathbb{R}} x_n f_{X_n}(x_n) dx_n \right\rangle \end{aligned}$$

where  $f_{X_i}$  is the marginal PDF of the random variable  $X_i$ .

**Definition 3.29 (Expected Value of a Function of a Random Vector)** Let  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  be a random vector with joint PDF  $f_{\mathbf{X}}(\cdot)$  and  $Y = g(X_1, \dots, X_n)$  a random variable. The expected value of  $Y$  is:

$$E[Y] = \int_{\mathbb{R}^n} g(x_1, \dots, x_n) f_{\mathbf{X}}(x_1, \dots, x_n) dx_1 \dots dx_n$$

Intuitively, the covariance matrix generalizes the notion of variance to multiple dimensions.

**Definition 3.30 (Covariance between Two Random Variables)** The covariance between two random variables  $X$  and  $Y$  with joint PDF  $f_{XY}(\cdot)$  is:

$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

**Property 3.11 (Computational Expression for the Covariance)** From the properties of the expected value:

$$\text{Cov}(X, Y) = E[XY] - E[X]E[Y]$$

**Definition 3.31 (Covariance Matrix)** Let  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  be a random vector with joint PDF  $f_{\mathbf{X}}(\cdot)$ . The covariance matrix  $\Sigma$  of  $\mathbf{X}$  is a matrix whose element in the  $i, j$  position is the covariance between its  $i^{\text{th}}$  and  $j^{\text{th}}$  elements:

$$\Sigma_{ij}[\mathbf{X}] = \text{Cov}(X_i, X_j)$$

Notice that each diagonal element  $\Sigma_{ii}$  is the variance of the  $i^{\text{th}}$  variable, i.e.,  $\Sigma_{ii}[\mathbf{X}] = \text{Cov}(X_i, X_i) = E[X_i X_i] - E[X_i]E[X_i] = \text{Var}[X_i]$ .

### 3.4.2 Conditioning

The probability of random vectors can be affected by the occurrence of some event on such random vector.

**Definition 3.32 (Conditional Joint Cumulative Distribution Function)**

Given a random vector  $\mathbf{X}$  on the probability space  $\langle \Omega, \mathcal{A}, P \rangle$  and a possible event  $B \in \mathcal{A}$ , (i.e.  $P(B) > 0$ ), the conditional joint CDF given  $B$  is defined by:

$$F_{\mathbf{X}|B}(\mathbf{x}) = P(\{\mathbf{X} \leq \mathbf{x}\}|B) = \frac{P(\{\mathbf{X} \leq \mathbf{x}\} \cap B)}{P(B)}, \quad \mathbf{x} \in \mathbb{R}^n$$

Similarly the conditional joint PDF can be defined for a random vector, given the evidence.

**Definition 3.33 (Conditional Joint Probability Density Function)** *Let*

$B \in \mathcal{A}$  be a possible event ( $P(B) > 0$ ),  $\mathbf{X}$  an  $n$ -dimensional random vector on the probability space  $\langle \Omega, \mathcal{A}, P \rangle$ , with conditional joint CDF given  $B$ ,  $F_{\mathbf{X}|B}(\cdot)$ , everywhere differentiable in every  $x_i$ , ( $1 \leq i \leq n$ ). The conditional joint PDF of  $\mathbf{X}$  given  $B$  is defined by:

$$f_{\mathbf{X}|B}(\mathbf{x}) = \frac{\partial^n F_{\mathbf{X}|B}(\mathbf{x})}{\partial x_1 \dots \partial x_n}$$

The conditional joint PDF can also be expressed directly in terms of the unconditional joint PDF.

**Property 3.12 (Alternative Expression for the Conditional PDF)**

Consider random vector  $\mathbf{X}$ , the set  $B' = \{\mathbf{X}(\omega) : \omega \in B\} \subseteq \mathbb{R}^n$  and its indicator function:

$$\mathbf{1}_{B'}(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in B' \\ 0, & \mathbf{x} \notin B' \end{cases}$$

An alternative expression for the conditional joint PDF of  $\mathbf{X}$  given  $B$  is

$$f_{\mathbf{X}|B}(\mathbf{x}) = \frac{f_{\mathbf{X}}(\mathbf{x})\mathbf{1}_{B'}(\mathbf{x})}{P(B)}$$

Thus,  $f_{\mathbf{X}|B}(\cdot)$  is proportional to  $f_{\mathbf{X}}(\cdot)$  on the set  $B'$ , and zero elsewhere.

The marginal CDFs and PDFs of each random variable  $X_i$  (with  $1 \leq i \leq n$ ) can be retrieved as defined in 3.23 and 3.26, respectively.

Similarly it is possible to define the conditional expected value of a random variable given an event.

**Definition 3.34 (Conditional Expected Value)** *Let  $B \in \mathcal{A}$  be an event with  $P(B) > 0$  and  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  be a random vector with conditional joint PDF given  $B$ ,  $f_{\mathbf{X}|B}(\cdot)$ . The conditional expected value of  $\mathbf{X}$  given  $B$  is:*

$$E[\mathbf{X}|B] = \langle E[X_1|B], \dots, E[X_n|B] \rangle = \left\langle \int_{\mathbb{R}} x_1 f_{X_1|B}(x_1) dx_1, \dots, \int_{\mathbb{R}} x_n f_{X_n|B}(x_n) dx_n \right\rangle$$

where  $f_{X_i|B}$  is the conditional marginal PDF of the random variable  $X_i$ .

**Definition 3.35 (Conditional Expected Value of a Function)** *Let  $B \in \mathcal{A}$  be an event with  $P(B) > 0$ ,  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  be a random vector with conditional joint PDF given  $B$ ,  $f_{\mathbf{X}|B}(\cdot)$ , and  $Y = g(\mathbf{X})$  a random variable. The conditional expected value of  $Y$  given  $B$  is:*

$$E[Y|B] = \int_{\mathbb{R}^n} g(\mathbf{x}) f_{\mathbf{X}|B}(\mathbf{x}) d\mathbf{x}$$

Since the conditional expected value  $E[\mathbf{X}|B]$  can be regarded as the mean of the conditional distribution of the random vector  $\mathbf{X}$  given  $B$ , all the properties of the unconditional expected value also apply.

Other conditional moments of a random variable, namely the conditional variance and the closely-related conditional standard deviation, are defined analogously.

## 3.5 Numerical Computations

Since many definitions in this chapter depend on the computation of multidimensional integrals, we introduce two alternative methods of computing the integral of a function over a box: the first, relying on Taylor models, provides a safe enclosure for the exact

integral value; the second, based on Monte Carlo integration techniques, provides only approximate enclosures, but is faster to execute.

These methods will be used in the next chapter to calculate enclosures for the probability of events and also for expected values and variances of random vectors.

### 3.5.1 Probabilistic Framework Outline

For problems defined over the  $n$ -dimensional Euclidean space,  $\mathbb{R}^n$  (or some subset of it), we adopt  $\langle \mathbb{R}^n, \mathcal{B}^n \rangle$  as the measurable space, where the sample space is an  $n$ -dimensional Euclidean space and  $\mathcal{B}^n$  is the  $n$ -dimensional Borel  $\sigma$ -algebra.

This sample space is included in  $\mathbb{R}^n$  and the original events coincide with the events induced via the random variables, so the identity random vector (see definition 3.21) is used. Hence, an event  $H \in \mathcal{B}^n$  and the event  $H_{\mathbf{X}}$  induced via the identity random vector  $X$  on  $H \subseteq \mathbb{R}^n$  are identical, i.e.,  $H_{\mathbf{X}} = \{\omega \in \Omega : \mathbf{X}(\omega) \in H\} = H$ .

Moreover, a probability measure  $P(\cdot)$  in the measurable space  $\langle \mathbb{R}^n, \mathcal{B}^n \rangle$  is based on the joint PDF,  $f_{\mathbf{X}}(\cdot)$ , of the random vector  $\mathbf{X}$ . Therefore, for any event  $H \in \mathcal{B}^n$ , the probabilistic measure is given by  $P(H) = \int_H f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad \mathbf{x} \in \mathbb{R}^n$ .

Summarizing, the probabilistic model considered in our framework is characterized by the following elements:

- Measurable space:  $\langle \mathbb{R}^n, \mathcal{B}^n \rangle$ ;
- Random vector: identity random vector  $\mathbf{X}$  with joint PDF  $f_{\mathbf{X}}(\cdot)$ ;
- Probability measure:  $P(H) = \int_H f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad \mathbf{x} \in \mathbb{R}^n$ ;
- Probability space:  $\langle \mathbb{R}^n, \mathcal{B}^n, P \rangle$

### 3.5.2 Integration with Taylor Models

In numerical analysis, the term quadrature is a synonym for numerical integration and several algorithms exist for calculating the numerical value of a definite integral.

The basic problem considered by numerical integration is to compute an approximate solution to a definite integral

$$S = \int_a^b f(x) dx.$$

If  $f$  is a smooth well-behaved function, integrated over a small number of dimensions and the limits of integration are bounded, there are many methods of approximating the integral. Nevertheless precise error bounds are rarely available. The error estimates, which are sometimes delivered, are not guaranteed and are sometimes unreliable.

Interval analysis techniques provide several methods [37, 71, 114] for unidimensional quadrature. These methods can be useful in our framework whenever the random variables of the probabilistic model are independent. In this case the joint PDF is the product of individual PDFs (see definition 3.27) and computing the probability of an event reduces to quadrature enclosures of univariate functions.

Methods to compute quadrature enclosures for multivariate functions are nevertheless needed when independence is not assumed on the random variables of the probabilistic model and a single joint PDF is provided. Quadrature methods can be based on Taylor models, (see definition 2.10). The Taylor model used here is slightly different from that presented in §2.1.2, since we will center the error component. To do so, in the adapted Taylor model of a function  $f$  in a box  $B$ , the polynomial part is defined by

$$p(x) = f(\tilde{x}) + \sum_{|\alpha|=1}^n \frac{1}{\alpha!} \frac{\partial^\alpha f(\tilde{x})}{\partial x^\alpha} (x - \tilde{x})^\alpha + \sum_{|\alpha|=n+1} c_\alpha (x - \tilde{x})^\alpha \quad (3.1)$$

and the remainder is defined by

$$R = \sum_{|\alpha|=n+1} ([r_\alpha](B) - c_\alpha)(B - \tilde{x})^\alpha \quad (3.2)$$

with  $[r_\alpha](B) = \frac{1}{\alpha!} \left[ \frac{\partial^\alpha f}{\partial x^\alpha} \right](B)$ ,  $c_\alpha = \text{mid}([r_\alpha](B))$  and  $\tilde{x} = \text{mid}(B)$ .

**Property 3.13 (Equivalent Inclusion Functions)** *The formula presented in definition 2.11 is equivalent to the inclusion function of the Taylor model that centers the error component (presented above).*

**Proof.** For the error-centered formula we have  $[f](B) = [p](B) + R$ , with  $p$  and  $R$ , defined above.  $R$  can be rewritten as

$$R = \sum_{|\alpha|=n+1} [r_\alpha](B) (B - \tilde{x})^\alpha - \sum_{|\alpha|=n+1} c_\alpha (B - \tilde{x})^\alpha$$

and the inclusion function of  $f$  becomes

$$[f](B) = f(\tilde{x}) + \sum_{|\alpha|=1}^n \frac{1}{\alpha!} \frac{\partial^\alpha f(\tilde{x})}{\partial x^\alpha} (B - \tilde{x})^\alpha + \sum_{|\alpha|=n+1} c_\alpha (B - \tilde{x})^\alpha + \sum_{|\alpha|=n+1} [r_\alpha](B) (B - \tilde{x})^\alpha - \sum_{|\alpha|=n+1} c_\alpha (B - \tilde{x})^\alpha.$$

Since the third and the last terms cancel we obtain the formula of definition 2.11. ■

The computation of an enclosure,  $[I]^*$ , for the quadrature of a multivariate function  $f$  is justified by the following lemma used in [17] and proved here for completeness.

**Lemma 3.1** *Let  $f(x) \in p(x) + R$  for all  $x$  in a given box  $B \in \mathbb{IR}^m$ . Then*

$$\int_B f(x) dx \in [I]^*(f, B) = \int_B p(x) dx + R \text{ vol}(B)$$

**Proof.** Since  $p(x) + \underline{R} \leq f(x) \leq p(x) + \overline{R}$ ,  $\forall x \in B$ , it is

$$\int_B p(x) + \underline{R} dx \leq \int_B f(x) dx \leq \int_B p(x) + \overline{R} dx$$

Since  $\int_B k dx = k \text{ vol}(B)$  for any constant  $k$ , we obtain

$$\int_B p(x) dx + \underline{R} \text{ vol}(B) \leq \int_B f(x) dx \leq \int_B p(x) dx + \overline{R} \text{ vol}(B)$$

and so the statement holds. ■

The enclosure provided by lemma 3.1 can be very sharp. Its computation reduces to the computation of the quadrature  $\int_B p(x) dx$  and of the remainder  $R$ .

**Property 3.14** *The quadrature  $\int_B p(x) dx$  can be computed by the analytic formula:*

$$\int_B p(x) dx = \text{vol}(B) f(\tilde{x}) + \sum_{|\alpha|=1}^n K_\alpha \int_B (x - \tilde{x})^\alpha dx + \sum_{|\alpha|=n+1} c_\alpha \int_B (x - \tilde{x})^\alpha dx$$

where  $K_\alpha = \frac{1}{\alpha!} \frac{\partial^\alpha f(\tilde{x})}{\partial x^\alpha}$  and  $\int_B (x - \tilde{x})^\alpha dx$  is given by property 3.15.

**Proof.** The quadrature  $\int_B p(x)dx$  in a box  $B = B_1 \times \cdots \times B_m$  can be expanded as

$$\begin{aligned} \int_B p(x)dx &= \int_B f(\tilde{x})dx + \int_B \sum_{|\alpha|=1}^n K_\alpha (x - \tilde{x})^\alpha dx + \int_B \sum_{|\alpha|=n+1} c_\alpha (x - \tilde{x})^\alpha dx \\ &= \text{vol}(B)f(\tilde{x}) + \sum_{|\alpha|=1}^n K_\alpha \int_B (x - \tilde{x})^\alpha dx + \sum_{|\alpha|=n+1} c_\alpha \int_B (x - \tilde{x})^\alpha dx \end{aligned}$$

where  $K_\alpha = \frac{1}{\alpha!} \frac{\partial^\alpha f(\tilde{x})}{\partial x^\alpha}$  is a constant. ■

The formula for  $\int_B (x - \tilde{x})^\alpha dx$  is now addressed.

**Property 3.15** *The analytic formula to compute  $\int_B (x - \tilde{x})^\alpha dx$ , where  $B = B_1 \times \cdots \times B_m$  and  $\tilde{x} = \text{mid}(B)$ , is given by:*

$$\int_B (x - \tilde{x})^\alpha dx = \begin{cases} 0 & \text{if } \exists \alpha_i \in \alpha \text{ odd}(\alpha_i) \\ \prod_{i=1}^m \frac{(w_i)^{\alpha_i+1}}{2^{\alpha_i}(\alpha_i + 1)} & \text{otherwise} \end{cases}$$

where  $w_i = \overline{B_i} - \underline{B_i}$ .

**Proof.**

$$\int_B (x - \tilde{x})^\alpha dx = \prod_{i=1}^m \int_{B_i} (x_i - \tilde{x}_i)^{\alpha_i} dx_i = \prod_{i=1}^m \frac{(x_i - \tilde{x}_i)^{\alpha_i+1} \Big|_{\underline{B_i}}^{\overline{B_i}}}{\alpha_i + 1}$$

This computation is simplified since we consider  $\tilde{x} = \text{mid}(B)$ . Focusing on the  $i^{\text{th}}$  component of the product, we have

$$\begin{aligned} \frac{(x_i - \tilde{x}_i)^{\alpha_i+1} \Big|_{\underline{B_i}}^{\overline{B_i}}}{\alpha_i + 1} &= \frac{\left(\overline{B_i} - \frac{\underline{B_i} + \overline{B_i}}{2}\right)^{\alpha_i+1}}{\alpha_i + 1} - \frac{\left(\underline{B_i} - \frac{\underline{B_i} + \overline{B_i}}{2}\right)^{\alpha_i+1}}{\alpha_i + 1} = \\ &= \frac{\left(\frac{\overline{B_i} - \underline{B_i}}{2}\right)^{\alpha_i+1}}{\alpha_i + 1} - \frac{\left(-\frac{\overline{B_i} - \underline{B_i}}{2}\right)^{\alpha_i+1}}{\alpha_i + 1} \end{aligned}$$

Denoting  $\overline{B}_i - \underline{B}_i$  as  $w_i$ , the previous formula can be further simplified to

$$\frac{(w_i)^{\alpha_i+1} - (-1)^{\alpha_i+1}(w_i)^{\alpha_i+1}}{2^{(\alpha_i+1)}(\alpha_i + 1)} = \begin{cases} 0 & \text{if } \text{odd}(\alpha_i) \\ \frac{(w_i)^{\alpha_i+1}}{2^{\alpha_i}(\alpha_i + 1)} & \text{if } \text{even}(\alpha_i) \end{cases}$$

It is then enough to have a single odd  $\alpha_i$  for the product to be zero. Otherwise it is the product of non-zero terms. So property 3.15 holds. ■

In practice, computing  $\int_B p(x)dx$  can only produce an approximate value, due to floating-point arithmetic rounding errors. By using interval arithmetic a rigorous enclosure of the scalar value can be maintained.

The computation of Taylor models of arbitrary expansion orders for multivariate functions, both for the polynomial part and for the remainder  $R$ , require the evaluation of higher order derivative tensors performed in interval arithmetic. The few available tools for computing higher order partial derivatives [30, 54] are based on floating-point arithmetic and, thus, are inadequate for our needs.

In the work developed with Goldztein and Cruz [48], we adopted an implementation of the recursive calculation in the forward mode of the chain-rule based technique know as automatic differentiation [18]. The idea is to pre-compile the integrand expression into a program code that computes all the derivative enclosures wrt to a box required by the Taylor model, and retrieves them, as needed. When computing the polynomial part, the box is an infinitesimal box around the midpoint of the domain of integration. When computing the remainder  $R$  the box is the domain of integration itself.

**Property 3.16 (Order of Convergence)** *The quadrature computed as in lemma 3.1 has an order of convergence  $n + 2 + m$  in a box  $B \in \mathbb{IR}^m$ , when an  $n$ -order Taylor model is used.*

**Proof.** See appendix B, section B.1. ■

In fact, when computing the integral of  $f$  inside a large box  $B$ ,  $R$  can be very wide (independently of the Taylor model order). To guarantee that the obtained enclosure is adequate we proceed as follows.

**Definition 3.36 (Safe Enclosure for the Integral of  $f$  over  $B$ )** Let  $f(x) \in p(x) + R$  for all  $x$  in  $B \in \mathbb{IR}^m$  and  $[f]$  a convergent inclusion function inside  $B$ :

$$\int_B f(x)dx \in [I](f, B) = [I]^*(f, B) \cap [f](B)vol(B)$$

where  $[I]^*$  is computed as in lemma 3.1.

Since the bound on the Taylor expansion improves as the box size decreases applying Taylor model based integration to sub-boxes whose union is the original box leads to a tighter enclosure of the initial integral [17, 93].

**Property 3.17 (Tighter Enclosure for the Integral of  $f$  over  $B$ )** Consider a box  $B$  and an almost disjoint set of boxes  $\{B_i : 1 \leq i \leq n\}$  such that  $\cup_{i=1}^n B_i = B$ .

Then  $\int_B f(x)dx \in \sum_{i=1}^n [I](f, B_i)$  and  $wid([I](f, B)) \geq wid\left(\sum_{i=1}^n [I](f, B_i)\right)$ .

We illustrate the previous computations by means of an example.

**Example 3.2.** Consider the standard bivariate normal PDF, with correlation coefficient  $\rho = 0.5$ , given by:

$$f(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right)$$

Using the method based on Taylor models to compute the enclosure for the quadrature of this function in the box  $B = [0, 0.5] \times [0, 0.5]$  (see Lemma 3.1) and adopting the Taylor model of order  $n = 2$  around the midpoint of  $B$ ,  $\tilde{x} = \langle 0.25, 0.25 \rangle$ , we obtain the enclosure for the quadrature of  $f$  in  $B$  (see appendix B, section B.2, for the intermediate computations):

$$\begin{aligned} \int_B f(x)dx &\in \int_B p(x)dx + R vol(B) = \\ &0.042870434037233 \left(\frac{9}{7}\right) + 0.0022615212900009 \times [-1, 1] = \\ &[0.0406089127472329, 0.0451319553272347]. \end{aligned}$$

To illustrate the convergence of the method based on Taylor models, table 3.1 shows the enclosures for the quadrature of  $f$  obtained with increasing orders ( $n = 2$ ,  $n = 5$ ,  $n = 10$ ,  $n = 15$  and  $n = 20$ ), around the midpoint of  $B$ .

$n$	Quadrature enclosure
2	0.04[06089127472329, 51319553272347]
5	0.042[8745381053786, 9201526531358]
10	0.04289729[31109425, 138828026]
15	0.042897298499[1777, 7946]
20	0.042897298499486[0, 6]

**Table 3.1:** Enclosure for the quadrature of  $f$  in  $B$  by Taylor models integration with increasing order  $n$ .

---

### 3.5.3 Integration with Monte Carlo

Deterministic numerical integration algorithms are well suited for a small number of dimensions, but difficulties arise in multivariate functions of high dimensions, both because the number of required function evaluations increases rapidly with the number of dimensions, and because the boundary of a multidimensional region may be highly non-linear making the reduction of the problem to a series of nested one-dimensional integrals infeasible.

Monte Carlo methods [56] provide an alternative approach to estimate the value of definite multidimensional integrals, with high dimensions. As long as the function is reasonably well-behaved, the integral can be estimated by randomly selecting  $N$  points in the multidimensional space and averaging the function value on these points. This method displays  $\frac{1}{\sqrt{N}}$  convergence, i.e., by quadrupling the number of sampled points the error is halved, regardless of the number of dimensions.

**Definition 3.37 (Monte Carlo Integration)** *Let  $H \subseteq \mathbb{R}^n$  be a region with a possible non-linear boundary,  $B$  an  $n$ -dimensional box and  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  a function. Consider  $N$  random sample points uniformly distributed inside  $B$ ,  $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  and*

the indicator function  $1_H$  defined as:

$$1_H(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \in (H \cap B) \\ 0 & \text{otherwise} \end{cases}$$

The average of  $f_H$  on the  $N$  sample points is:

$$\langle 1_H \rangle = \frac{\sum_{i=1}^N 1_H(\mathbf{x}_i)}{N}$$

Monte Carlo integration estimates the integral of  $f$  over  $(H \cap B)$  with:

$$\int_B 1_H(\mathbf{x}) d\mathbf{x} = \int_{(H \cap B)} f(\mathbf{x}) d\mathbf{x} \approx \hat{I}_H(f, B) = \langle 1_H \rangle \text{vol}(B)$$

**Property 3.18 (Convergence)** *By the law of large numbers Monte Carlo estimate converges to the true value of the integral:*

$$\lim_{N \rightarrow \infty} \hat{I}_H(f, B) = \int_{(H \cap B)} f(\mathbf{x}) d\mathbf{x}$$

Besides obtaining an estimate of the integral value (by the law of large numbers), Monte Carlo integration provides an estimate of the uncertainty in the estimate (by the central limit theorem). For more details see [70, section 2.7],[108, chapter 2].

**Definition 3.38 (Standard Deviation of the Estimate)** *From the central limit theorem the standard deviation of the estimate of the integral,  $\hat{I}_H(f, B)$ , as calculated in 3.37 is:*

$$\sigma(\hat{I}_H(f, B)) = \text{vol}(B) \sqrt{\frac{\langle 1_H^2 \rangle - \langle 1_H \rangle^2}{N}}$$

where  $\langle 1_H^2 \rangle$  is the average of  $1_H^2$  on the  $N$  sample points:

$$\langle 1_H^2 \rangle = \frac{\sum_{i=1}^N (1_H(\mathbf{x}_i))^2}{N}$$

The standard deviation provides a statistical estimate of the error on the result given by Monte Carlo integration. By the central limit theorem, since the error is assumed to be normally distributed, this means that the probability that the true value is within one sigma error is about  $2/3$ .

The error estimate is not a strict error bound since random sampling of the region of interest may not uncover all the important features of the function, resulting in an underestimate of the error. Furthermore there is no guarantee that the error is normally distributed, so the error term should be taken only as an indicator.

**Property 3.19 (Order of Convergence)** *The central limit theorem ensures that Monte Carlo integration converges with order of  $\frac{1}{\sqrt{N}}$ .*

Since the order of convergence of Monte Carlo integration is decelerating much of the effort in the development of this method has been focused on variance reduction techniques [22, 108] (e.g. antithetic variables, control variates, importance sampling and stratified sampling).

Recursive stratified sampling is an adaptive method that estimates, on each step, the integral and the error using the basic Monte Carlo algorithm. If the error estimate is larger than a given accuracy the integration region is divided into subregions and the procedure is recursively applied to them. The estimate and its error are obtained by adding up the partial results. By choosing the subregion with the highest error estimate to process next this method concentrates sampling in regions of higher variance making it more effective. If the subregions and the number of samples in each subregion are adequately chosen then this method can lead to a significant variance reduction when compared with the basic Monte Carlo.

**Example 3.3.** Consider again function  $f$  of example 3.2. We use the basic Monte Carlo integration method to compute an approximate value for the quadrature of this function in box  $B = [0, 0.5] \times [0, 0.5]$  together with its standard deviation (in the example  $H = B$  and, as such, the domain of integration is  $B$ ). Three different values for  $N$  are used:  $N = 5$ ,  $N = 25$  and  $N = 125$  and the results are shown in table 3.2. The first column presents the number of sample points, the second presents the

---

$N$	$I_{MC}(B)$	$\sigma(I_{MC}(B))$	$[I_{MC}(B) - \sigma(I_{MC}(B)), I_{MC}(B) + \sigma(I_{MC}(B))]$
5	0.04194	0.00063	[0.04131, 0.04257]
25	0.04256	0.00032	[0.04224, 0.04288]
125	0.04304	0.00016	[0.04288, 0.0432]

**Table 3.2:** Estimated integral of  $f$  in  $B$  by Monte Carlo integration and correspondent error estimate.

approximate value obtained for the quadrature and the last shows the estimate for the error (standard deviation).

Although the results obtained agree with those obtained with Taylor models (see table 3.1) only when  $N = 125$  does the interval ([0.04288, 0.0432]) enclose the correct value. It is possible to observe the order of convergence of the method (the estimated error decreases by half when the number of samples is approximately quadrupled).

---

## 3.6 Summary

This chapter introduced relevant concepts of probability theory, presenting unconditional and conditional probabilities, random variables and random vectors, in a continuous probability space. In this context, two numerical integration methods were presented to compute either safe or approximate enclosures for exact integral values: Taylor Models and Monte Carlo methods, respectively.

The next part of the thesis presents the proposed probabilistic continuous constraint framework, showing how to combine continuous constraint programming and probability theory. In particular, the next chapter introduces the concept of probabilistic continuous constraint space and presents methods to compute safe and approximate enclosures for unconditional and conditional probabilities of events within such space.



## Part II

# Probabilistic Constraints



## CHAPTER 4

# Probabilistic Constraint Programming

Although the classical CSP framework is a powerful and expressive paradigm to represent many kinds of real world problems, the CSP research community early noticed that it had important limitations to cope with real world situations where the knowledge was not completely available nor crisp. Such situations had an impact on several aspects of a CSP, namely on the effective number of constraints to be satisfied, on the existence of preferences for the possible values of the variables, on the distinction between decision variables (for which the decision maker can set a value) and uncontrollable variables (that represent states of nature). All these aspects justified extensions of the CSP framework, namely for finite domain problems, each addressing one or more of these issues.

For example [110] formalizes a model that considers some variable values to be preferred over others, presenting such preferences by a probability distribution over the variables values. Other proposals aim to represent real world problems where it is not mandatory to satisfy all the constraints [41] or where a constraint has a certain probability of being part of the problem [43]. Fuzzy CSP, associates values to each tuple of constraints, or to each constraint, which indicate the level of preference for satisfying them. In [19] the authors propose a formalism, based on semirings, to unify previous approaches that associate preferences either to the values in the domain of a variable, or to sets of constraints. In [44, 117] the authors extend the CSP framework to deal with decision problems under uncertainty. A distinction is made between decision and uncontrollable variables, the latter representing knowledge about the world by a probability distribution of their values.

All the above approaches deal with discrete domains and although the ideas may be, in some cases, similar to those explored in this thesis, both the techniques and the modeled problems are necessarily different, since we consider continuous domains.

A combination of probabilistic and interval representations of uncertainty appears in [76]. This approach is specially suited for data processing problems, where an estimate for an output quantity is computed by applying a known deterministic algorithm to the available estimates for other quantities. Hence, intervals are maintained representing possible values of both variables and parametric descriptors of their distributions (e.g., expected values). Throughout interval propagation such intervals are maintained consistent by an evaluation process that extends basic interval arithmetic operations (see [76] for details). Contrary to constraint approaches, that are based on undirected relations, this approach is highly dependent on the availability of a directed algorithm to compute the intended information for the output variable from the input estimates, making it less general than constraint based paradigms.

Quantified CSPs [13] address the distinction between decision and uncontrollable variables. Uncontrollable variables are assumed to be universally quantified, and the goal is to find values for the decision variables that satisfy the problem constraints, for any possible values of the uncontrollable variables. Nevertheless this approach does not include probabilistic information.

The main contribution of the present thesis is the proposal of the Probabilistic Continuous Constraint Framework as an extension of the Continuous Constraint Satisfaction paradigm to complement the interval bounded representation of uncertainty with a probabilistic characterization of the distribution of values.

The proposed approach provides an extra characterization of uncertainty by considering a probabilistic space associated with a CCSP. By reasoning over this probabilistic space and considering adequate events on it, it is possible to compute the relevant probabilistic features, allowing decisions to be more suitably informed.

In this chapter the framework is defined and formalized and its main properties are highlighted. In section 4.1 the notion of Probabilistic Continuous Constraint Space is defined, based on the concepts presented in the previous chapters of this thesis. Section 4.2 presents the specific characteristics of the events handled by the Probabilistic Constraint framework and conceptually specifies how to compute the probability of such

events. Subsequently, section 4.3 specifies how to compute safe enclosures for the value of multidimensional integrals over regions defined by the events. Relying on safe integration methods and on continuous constraint reasoning, sections 4.4 and 4.5 propose techniques to compute safe enclosures for, respectively, the probability and conditional probability of events. Section 4.6 proposes algorithms to compute the enclosures discussed in the two previous sections and section 4.7 focuses on an alternative, if only approximate, method to compute enclosures for multidimensional integrals over events and discusses the impact of adopting such method. Finally, section 4.8 applies the proposed algorithms to a set of benchmarks and discusses the results obtained.

## 4.1 Probabilistic Continuous Constraint Space

A Probabilistic Continuous Constraint Space (hereafter referred as PC) associates a probabilistic space to a continuous constraint satisfaction problem. In the following a PC is defined incrementally, based on the definitions of sections 3.1, 3.5.1 and 2.2. Firstly, a probability space is associated with a CCSP.

**Definition 4.1 (PC Probability Space)** *Given a CCSP  $L = \langle X, D, C \rangle$ , the associated probability space is  $\langle \Omega, \mathcal{B}^n, P \rangle$  where  $\Omega \supseteq D \in \mathbb{I}\mathbb{R}^n$ ,  $\mathcal{B}^n$  is the  $n$ -dimensional Borel  $\sigma$ -algebra on  $\Omega$  and  $P$  is a probability measure.*

Secondly, the variables of the PC are mapped onto random variables.

**Definition 4.2 (PC Random Vector)** *Given a PC probability space on  $\langle \langle x_1, \dots, x_n \rangle, D, C \rangle$ , an identity random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  is considered, with joint PDF  $f(\cdot)$ . Each component random variable  $X_i : \mathbb{R}^n \rightarrow \mathbb{R}$  is defined as:*

$$X_i(\Omega) = \Pi_i[\Omega]$$

Thirdly, the probability measure  $P$  is defined.

**Definition 4.3 (PC Probability Measure)** *Given a probability space  $\langle \Omega, \mathcal{B}^n, P \rangle$  and the identity multivariate random variable  $\mathbf{X}$  with joint PDF  $f(\cdot)$ , the probability measure  $P$  that assigns a probability to any event  $H \in \mathcal{B}^n$  is defined as:*

$$P(H) = \int \dots \int_H f(x_1, \dots, x_n) dx_n \dots dx_1$$

Finally, a Probabilistic Continuous Constraint Space may be defined.

**Definition 4.4 (Probabilistic Continuous Constraint Space)** *A Probabilistic Continuous Constraint Space is a pair  $\langle \langle X, D, C \rangle, f \rangle$  where  $\langle X, D, C \rangle$  is a CCSP and  $f$  is the joint PDF of the identity multivariate random variable  $\mathbf{X}$  defined on the probability space associated with the PC.*

A PC defines a probabilistic model that encodes probabilistic information. In this context, a problem over a PC is defined as follows.

**Definition 4.5 (Probabilistic Continuous Constraint Problem)** *Given a PC  $\langle \langle X, D, C \rangle, f \rangle$ , a Probabilistic Continuous Constraint Problem is a query that requires the computation of probabilistic information over elements of the PC (PC events<sup>1</sup> or random variables).*

Given a PC, the following problems can be formulated in the PC framework:

- Probability of a PC event;
- Probability that a random vector takes a range of values;
- Expected value of a random vector;
- Covariance matrix of a random vector;
- Probability distribution of a random vector;
- Conditional version of the above problems, given a PC event.

---

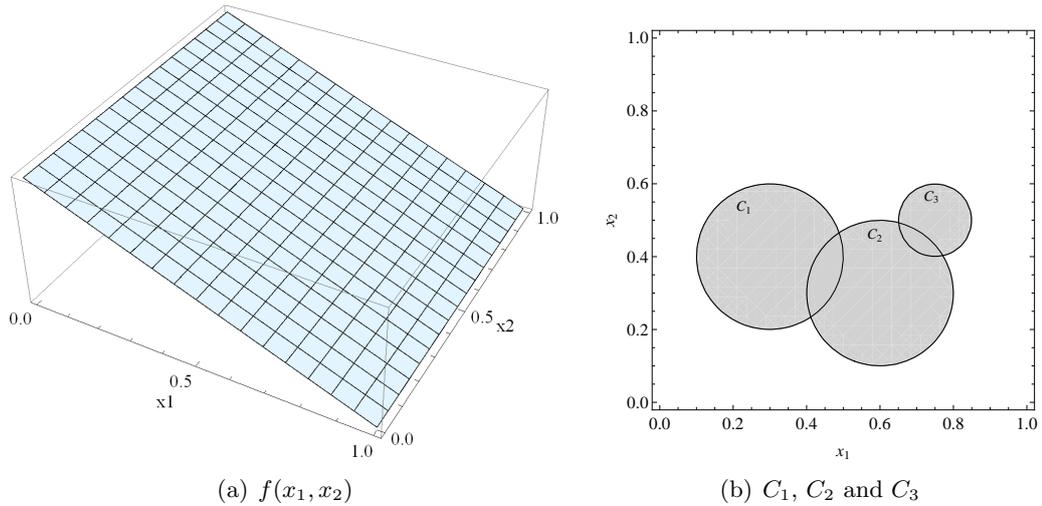
<sup>1</sup>See definition 4.2 in the next section.

Next we show an example of a PC.

**Example 4.1.** Consider PC  $Example = \langle \langle X = \langle x_1, x_2 \rangle, D = [0, 1] \times [0, 1], C \rangle, f \rangle$ , where  $f(x_1, x_2) = 2 - 2x_1$  (shown in figure 4.1 (a)),  $C = \{C_1 : (x_1 - 0.3)^2 + (x_2 - 0.4)^2 \leq 0.04, C_2 : (x_1 - 0.6)^2 + (x_2 - 0.3)^2 \leq 0.04, C_3 : (x_1 - 0.75)^2 + (x_2 - 0.5)^2 \leq 0.01\}$  (shown in figure 4.1 (b)),  $\Omega = D$  and the associated random vector  $\mathbf{X} = \langle X_1, X_2 \rangle$ .

As expected  $\int_{\Omega} f(x_1, x_2) dx_1 dx_2 = 1$ .

Notice that there are no solutions to the underlying CCSP when all constraints are imposed (as can be seen in figure 4.1 (b)). Nevertheless the formulation of PC problems is not affected in that constraints are associated to events and the problems of interest (e.g. querying the probability of events) involve a subset of these constraints.



**Figure 4.1:** (a) Joint PDF of  $\mathbf{X}$  (b) Constraints.

## 4.2 Probabilistic Constraint Events

Given a PC  $\langle \langle X, D, C \rangle, f \rangle$ , we are interested in calculating the probability of events. In this context an event can be a box  $B \subseteq D$ , a constraint  $(s, \rho) \in C$  or a conjunction of such events.

**Definition 4.6 (Probabilistic Constraint Event)** Given a PC  $P = \langle \langle X, D, C \rangle, f \rangle$ ,  $\mathcal{H}$  is a probabilistic constraint event wrt  $P$  iff it is the feasible space of a CCSP  $\langle X, D_{\mathcal{H}}, C_{\mathcal{H}} \rangle$ , where  $D_{\mathcal{H}} \subseteq D$  and  $C_{\mathcal{H}} \subseteq C$ , i.e.,  $\mathcal{H} = \mathcal{F}(\langle X, D_{\mathcal{H}}, C_{\mathcal{H}} \rangle)$ .

We will often consider two special cases of PC events: box events and constraint events. A box event is the feasible space of a CCSP with no constraints, i.e.,  $\mathcal{F}(\langle X, D_{\mathcal{H}}, \emptyset \rangle) = D_{\mathcal{H}}$ . A constraint event is the feasible space of a CCSP with a single constraint, i.e.,  $\mathcal{F}(\langle X, D_{\mathcal{H}}, C_{\mathcal{H}} \rangle)$  where  $D_{\mathcal{H}} = D$  and  $C_{\mathcal{H}} = \{(s, \rho)\}$ .

When there is no ambiguity wrt the CCSP to which a PC event is related, this will be denoted by  $\mathcal{H}$  (similarly to the  $\mathcal{F}$  used to denote the feasible space of a CCSP).

In this thesis we only consider inequality constraints defined by closed-form expressions (as in definition 2.13) and we assume that the corresponding PC events are in the Borel  $\sigma$ -algebra of  $\Omega$ .

**Example 4.2.** Consider PC *Example* (see example 4.1) and figure 4.2, that shows its three constraints ( $C_1$ ,  $C_2$  and  $C_3$ ) and five boxes ( $B_i \subseteq D$ , with  $1 \leq i \leq 5$ ).

The figure highlights four PC events obtained by conjunction of box and constraint events, namely:

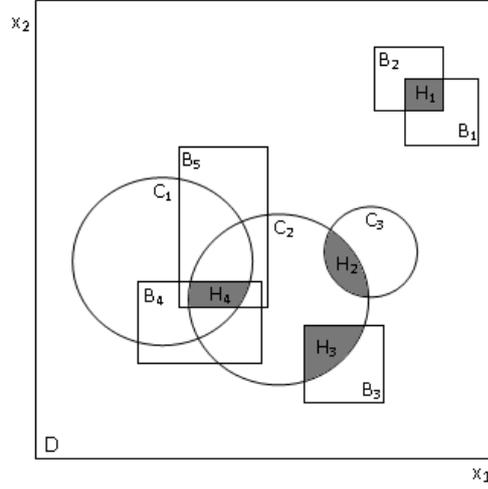
$$\begin{aligned} \mathcal{H}_1 &= \mathcal{F}(\langle X, B_1, \emptyset \rangle) \cap \mathcal{F}(\langle X, B_2, \emptyset \rangle) = \mathcal{F}(\langle X, B_1 \cap B_2, \emptyset \rangle) = B_1 \cap B_2 \\ \mathcal{H}_2 &= \mathcal{F}(\langle X, D, \{C_2\} \rangle) \cap \mathcal{F}(\langle X, D, \{C_3\} \rangle) = \mathcal{F}(\langle X, D, \{C_2, C_3\} \rangle) \\ \mathcal{H}_3 &= \mathcal{F}(\langle X, B_3, \emptyset \rangle) \cap \mathcal{F}(\langle X, D, \{C_2\} \rangle) = \mathcal{F}(\langle X, B_3, \{C_2\} \rangle) \\ \mathcal{H}_4 &= \mathcal{F}(\langle X, B_4, \emptyset \rangle) \cap \mathcal{F}(\langle X, B_5, \emptyset \rangle) \cap \mathcal{F}(\langle X, D, \{C_1\} \rangle) \cap \mathcal{F}(\langle X, D, \{C_2\} \rangle) \\ &= \mathcal{F}(\langle X, B_4 \cap B_5, \{C_1, C_2\} \rangle). \end{aligned}$$


---

The probability of a PC event  $\mathcal{H}$ , given a PDF  $f(\cdot)$ , is computed from the multidimensional integral of  $f(\cdot)$  on the region defined by the event.

**Definition 4.7 (Probability of a PC Event)** Given a PC  $\langle \langle X, D, C \rangle, f \rangle$  and a PC event  $\mathcal{H}$ , the probability of  $\mathcal{H}$  is defined as:

$$P(\mathcal{H}) = \int_{\mathcal{H}} f(\mathbf{x}) d\mathbf{x}$$



**Figure 4.2:** Some possible events of a probability space defined by a PC.

**Example 4.3.** Figure 4.3 (a) shows boxes  $B_i$  ( $1 \leq i \leq 4$ ) and constraint  $C_1$ , informally defined as  $x_2 \leq x_1$ . Consider PC  $\text{Triangle} = \langle \langle X, D, \{C_1\} \rangle, f \rangle$  (with  $X$ ,  $D$  and  $f$  defined as in the PC *Example*) and the following events:

$$\begin{aligned} \mathcal{H}_1 = B_1 &= \mathcal{F}(\langle X, [0, 0.5] \times [0.5, 1], \emptyset \rangle) & \mathcal{H}_2 = B_2 &= \mathcal{F}(\langle X, [0.5, 1] \times [0.5, 1], \emptyset \rangle) \\ \mathcal{H}_3 = B_3 &= \mathcal{F}(\langle X, [0, 0.5] \times [0, 0.5], \emptyset \rangle) & \mathcal{H}_4 = B_4 &= \mathcal{F}(\langle X, [0.5, 1] \times [0, 0.5], \emptyset \rangle) \\ \mathcal{H}_5 &= \mathcal{F}(\langle X, D, \{C_1\} \rangle) & \mathcal{H}_6 = D &= \mathcal{F}(\langle X, D, \emptyset \rangle) \end{aligned}$$

From definition 4.7, the probability of a box event  $\mathcal{H} = [a, b] \times [c, d]$  is:

$$P(\mathcal{H}) = \int_a^b \int_c^d f(x_1, x_2) dx_2 dx_1 = (2b - b^2 - 2a + a^2)(d - c) \quad (4.1)$$

So,  $P(\mathcal{H}_1) = \frac{3}{8}$ ,  $P(\mathcal{H}_2) = \frac{1}{8}$ ,  $P(\mathcal{H}_3) = \frac{3}{8}$ ,  $P(\mathcal{H}_4) = \frac{1}{8}$ , and, as expected,  $P(\mathcal{H}_6) = 1$ .

The probability of the constraint event  $\mathcal{H}_5$  can be obtained analytically:

$$P(\mathcal{H}_5) = \int_0^1 \int_0^{x_1} f(x_1, x_2) dx_2 dx_1 = \frac{1}{3}$$

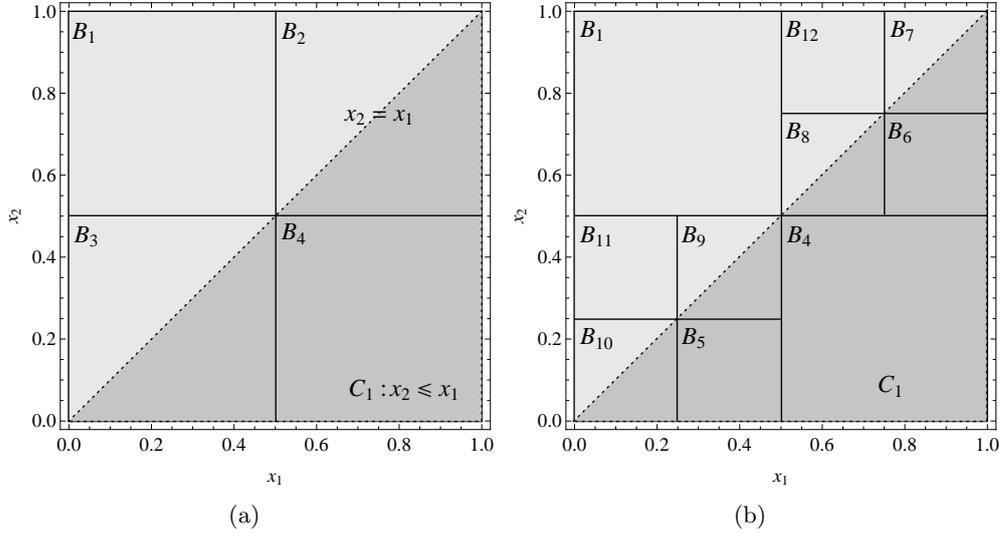


Figure 4.3: Boxes and constraint

In the following we will use concepts from §2.3.3<sup>1</sup>.

Since a PC event is the feasible space of a CCSP, an outer box cover of a PC event (see definition 2.27) can be used to define its probability.

**Property 4.1 (Probability of a PC Event through an Outer Box Cover)**

Given a PC event  $\mathcal{H}$  and an outer box cover  $\mathcal{H}_\square = \{B_1, \dots, B_n\}$  of  $\mathcal{H}$ , the probability of  $\mathcal{H}$  is given by:

$$P(\mathcal{H}) = \sum_{B \in \mathcal{H}_\square} P(B \cap \mathcal{H})$$

In the following  $P_{\mathcal{H}}(B)$  will denote  $P(B \cap \mathcal{H})$ , where  $B \in \mathcal{H}_\square$ .

**Proof.** Since  $\mathcal{H}_\square$  is an almost disjoint set of boxes then  $S_{\mathcal{H}} = \{B_i \cap \mathcal{H} : 1 \leq i \leq n\}$  is an almost disjoint set of events. Given that  $\mathcal{H} = \bigcup S_{\mathcal{H}}$ , by property 3.3,  $P(\mathcal{H}) = P(\bigcup S_{\mathcal{H}}) = \sum_{H_i \in S_{\mathcal{H}}} P(H_i)$ . ■

<sup>1</sup>Given an event  $\mathcal{H}$  and a joint box cover  $\mathcal{H}_\boxplus = \langle \mathcal{H}_\square, \mathcal{H}_\blacksquare \rangle$  of  $\mathcal{H}$ , then  $\mathcal{H}_\square$  is an outer box cover of  $\mathcal{H}$  and  $\mathcal{H}^+ = \bigcup \mathcal{H}_\square$  is an outer approximation of  $\mathcal{H}$ ;  $\mathcal{H}_\blacksquare \subseteq \mathcal{H}_\square$  is an inner box cover of  $\mathcal{H}$  and  $\mathcal{H}^- = \bigcup \mathcal{H}_\blacksquare$  is an inner approximation of  $\mathcal{H}$ ; and  $\Delta\mathcal{H} = \bigcup (\mathcal{H}_\square \setminus \mathcal{H}_\blacksquare)$  is a boundary approximation of  $\mathcal{H}$ .  $\mathcal{H}_\square$  and  $\mathcal{H}_\blacksquare$  are, by definition, mutually almost disjoint sets of boxes.

This property is illustrated in our working example.

**Example 4.4.** Consider the PC *Triangle* (example 4.3) and the boxes shown in figure 4.3 (b).

$$\begin{aligned}
 B_1 &= [0, 0.5] \times [0.5, 1] & B_4 &= [0.5, 1] \times [0, 0.5] \\
 B_5 &= [0.25, 0.5] \times [0, 0.25] & B_6 &= [0.75, 1] \times [0.5, 0.75] \\
 B_7 &= [0.75, 1] \times [0.75, 1] & B_8 &= [0.5, 0.75] \times [0.5, 0.75] \\
 B_9 &= [0.25, 0.5] \times [0.25, 0.5] & B_{10} &= [0, 0.25] \times [0, 0.25] \\
 B_{11} &= [0, 0.25] \times [0.25, 0.5] & B_{12} &= [0.5, 0.75] \times [0.75, 1]
 \end{aligned}$$

The probability of the conjunction of a box event  $B = [a, b] \times [a, b]$  (*diagonal box*) with the constraint event  $\mathcal{H}_5$  is given by:

$$P(B \cap \mathcal{H}_5) = \int_a^b \int_a^{x_1} (2 - 2x_1) dx_2 dx_1 = -2ab + b^2 + ab^2 - \frac{2}{3}b^3 + a^2 - \frac{1}{3}a^3 \quad (4.2)$$

Table 4.1 shows the probabilities of each  $B_i$  ( $4 \leq i \leq 10$ ) using formula (4.1) and of the conjunction of  $B_i$  with  $\mathcal{H}_5$ , using formulas (4.1) and (4.2) as adequate.

$B$	$P(B)$	$P_{\mathcal{H}_5}(B)$
$B_1 = \mathcal{H}_1$	3/8	0
$B_2 = \mathcal{H}_2$	1/8	1/24
$B_3 = \mathcal{H}_3$	3/8	1/6
$B_4 = \mathcal{H}_4$	1/8	1/8
$B_5$	5/64	15/192
$B_6$	1/64	3/192
$B_7$	1/64	1/192
$B_8$	3/64	4/192
$B_9$	5/64	7/192
$B_{10}$	7/64	10/192

**Table 4.1:** Probabilities of  $B_i$  and  $B_i \cap \mathcal{H}_5$  using formulas (4.1) and (4.2).

Let  $\mathcal{H}_\square = \{B_i : 4 \leq i \leq 10\}$  be an outer box cover of event  $\mathcal{H}_5$ . By definition 4.1, the probability of  $\mathcal{H}_5$  is given by  $\sum_{i=4}^{10} P_{\mathcal{H}_5}(B_i)$ . Using the values of table 4.1 we obtain  $P(\mathcal{H}_5) = \frac{64}{192} = \frac{1}{3}$ . As expected, this result is exactly the same as that obtained when the box cover was not used (in example 4.3).

---

### 4.3 Safe Integral Enclosure

Definition 4.7 and property 4.1 can rarely be applied to arbitrary PC events. Unlike the examples presented in the previous section, the multidimensional integral cannot be easily computed, in general, since (1) the definite integral of  $f$  may have no closed-form solution and/or (2) the event  $\mathcal{H}$  may establish a complex nonlinear integration boundary.

To cope with (1) two approaches can be adopted: (a) compute a safe enclosure  $[I_{\mathcal{H}}](f, B)$  for the integral of  $f$  over an event  $\mathcal{H}$  within an adequate box  $B$ , i.e.,  $\int_{B \cap \mathcal{H}} f(\mathbf{x}) d\mathbf{x} \in [I_{\mathcal{H}}](f, B)$ ; or (b) compute an approximate enclosure  $\widehat{[I_{\mathcal{H}}]}(f, B)$ , i.e.,  $\int_{B \cap \mathcal{H}} f(\mathbf{x}) d\mathbf{x} \simeq \widehat{[I_{\mathcal{H}}]}(f, B)$ . All definitions and properties presented in this chapter assume safe enclosures, except in section 4.7 that proposes a method to compute approximate enclosures and discusses the impact of adopting such approximate enclosures.

Section 3.5.2 presented a method based on Taylor models that can be used to compute safe enclosures for the integral of a function over a box. To compute an integral of a function over some region defined as a box, definition 3.36 can be applied to obtain a sharp enclosure. However, when the region is some unknown subset of the box (eventually empty) a cruder enclosure can be used, ranging from zero to the integral of the function maximum (minimum) over the entire box.

**Property 4.2 (Safe Enclosure for the Integral of  $h$  over (Box  $\cap$  PC Event))**

Given a joint box cover  $\mathcal{H}_{\boxplus} = \langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle$  of a PC event  $\mathcal{H}$  and a Taylor model  $\langle p, R \rangle$  of a function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  inside an  $n$ -dimensional box  $B \in \mathcal{H}_{\square}$ :

$$\int_{B \cap \mathcal{H}} h(\mathbf{x}) d\mathbf{x} \in [I_{\mathcal{H}}](h, B) = \begin{cases} [I](h, B) & \text{if } B \in \mathcal{H}_{\blacksquare} \\ [0] \uplus [h](B) \text{ vol}(B) & \text{otherwise} \end{cases}$$

where  $[I]$  is computed as in 3.36 and  $[h]$  is a convergent inclusion function of  $h$ .

**Proof.** If  $B \in \mathcal{H}_{\blacksquare}$  then  $(B \cap \mathcal{H}) = B$  and  $\int_{(B \cap \mathcal{H})} h(\mathbf{x}) d\mathbf{x} = \int_B h(\mathbf{x}) d\mathbf{x} \in [I](h, B)$ .

If  $B \notin \mathcal{H}_\blacksquare$ , let  $I = [h](B)$  and, consequently,  $\forall_{\mathbf{x} \in B} \underline{I} \leq h(\mathbf{x}) \leq \bar{I}$  then

$$\underline{I} \text{vol}(B \cap \mathcal{H}) \leq \int_{(B \cap \mathcal{H})} h(\mathbf{x}) d\mathbf{x} \leq \bar{I} \text{vol}(B \cap \mathcal{H}) \Leftrightarrow \quad (4.3)$$

$$\min(0, \underline{I}) \text{vol}(B) \leq \int_{(B \cap \mathcal{H})} h(\mathbf{x}) d\mathbf{x} \leq \max(0, \bar{I}) \text{vol}(B) \Leftrightarrow \quad (4.4)$$

$$\int_{(B \cap \mathcal{H})} h(\mathbf{x}) d\mathbf{x} \in [0] \uplus \text{vol}(B)[h](B) \quad (4.5)$$

Transition from (4.3) to (4.4) is based on the fact that  $\text{vol}(B \cap \mathcal{H}) \leq \text{vol}(B)$  and so  $\min(0, \underline{I}) \text{vol}(B) \leq \underline{I} \text{vol}(B \cap \mathcal{H})$  and  $\bar{I} \text{vol}(B \cap \mathcal{H}) \leq \max(0, \bar{I}) \text{vol}(B)$ . Transition from (4.4) to (4.5) comes from equality  $[\min(0, \underline{I}), \max(0, \bar{I})] = [0] \uplus I = [0] \uplus [h](B)$ . ■

To deal with the complex nonlinear integration boundary of a PC event  $\mathcal{H}$  and obtain guaranteed enclosures for the exact integral of  $\mathcal{H}$ , we can use property 4.2, as follows.

**Property 4.3 (Enclosure for the Integral of  $h$  over a PC Event)** *Given a joint box cover  $\mathcal{H}_\boxplus = \langle \mathcal{H}_\square, \mathcal{H}_\blacksquare \rangle$  of a PC event  $\mathcal{H}$ , an enclosure for the integral of a function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  in  $\mathcal{H}$  is given by:*

$$\int_{\mathcal{H}} h(\mathbf{x}) d\mathbf{x} \in [I](\mathcal{H}_\boxplus, h) = \sum_{B \in \mathcal{H}_\square} [I_{\mathcal{H}}](h, B)$$

**Proof.** Given the almost disjoint set of boxes  $\mathcal{H}_\square$  then  $\{B \cap \mathcal{H} : B \in \mathcal{H}_\square\}$  is a partition<sup>1</sup> of  $\mathcal{H}$ . Since integrals are additive with respect to the partitioning of the integration region into pairwise disjoint regions then

$$\int_{\mathcal{H}} h(\mathbf{x}) d\mathbf{x} = \sum_{B \in \mathcal{H}_\square} \int_{B \cap \mathcal{H}} h(\mathbf{x}) d\mathbf{x}$$

Given an enclosure for  $\int_{B \cap \mathcal{H}} h(\mathbf{x}) d\mathbf{x}$ , as computed in property 4.2, by the properties of interval arithmetic :

$$\int_{\mathcal{H}} h(\mathbf{x}) d\mathbf{x} \in \sum_{B \in \mathcal{H}_\square} [I_{\mathcal{H}}](h, B) = [I](\mathcal{H}_\boxplus, h) \quad \blacksquare$$

<sup>1</sup>By definition, the sets of a partition  $P$  are pairwise disjoint. However, in this case, they are

## 4.4 Probability Enclosure

All the definitions and properties that follow consider, implicitly, an underlying PC  $\langle\langle X, D, C \rangle, f\rangle$ .

When computing the probability of an event, the integrand function,  $h = f$ , is positive and property 4.2 can be presented in a simplified way.

### Property 4.4 (Enclosure for the Probability of (Box $\cap$ PC Event))

Consider a PC event  $\mathcal{H}$  and a joint box cover  $\langle\mathcal{H}_\square, \mathcal{H}_\blacksquare\rangle$  of  $\mathcal{H}$ . An enclosure for the probability of  $\mathcal{H} \cap B$ , for any box  $B \in \mathcal{H}_\square$  is given by:

$$\forall_{B \in \mathcal{H}_\square} P_{\mathcal{H}}(B) \in [P_{\mathcal{H}}](f, B) = \begin{cases} [I](f, B) & \text{if } B \in \mathcal{H}_\blacksquare \\ [0, \sup([f](B)) \text{ vol}(B)] & \text{otherwise} \end{cases}$$

To obtain guaranteed enclosures for the exact probability of a PC event  $\mathcal{H}$ , we adapt property 4.3 to the probability enclosure of  $\mathcal{H}$  within a box  $B$ , as follows.

### Property 4.5 (Enclosure for the Probability of a PC Event)

Consider a PC event  $\mathcal{H}$  and a joint box cover  $\mathcal{H}_{\boxplus} = \langle\mathcal{H}_\square, \mathcal{H}_\blacksquare\rangle$  of  $\mathcal{H}$ . An enclosure for the probability of  $\mathcal{H}$  is given by:

$$P(\mathcal{H}) \in [P](\mathcal{H}_{\boxplus}, f) = \sum_{B \in \mathcal{H}_\square} [P_{\mathcal{H}}](f, B)$$

**Proof.** Similar to the proof of property 4.3. ■

These property is illustrated in our working example.

**Example 4.5.** Consider PC *Triangle* data (from example 4.3), the constraint event  $\mathcal{H}_5$  (with  $P(\mathcal{H}_5) = \frac{1}{3}$ ) and its joint box cover  $\mathcal{H}_{\boxplus_1} = \langle\{B_2, B_3, B_4\}, \{B_4\}\rangle$ . Table 4.2 shows enclosures for the probabilities of the conjunction of each  $B_i$  with  $\mathcal{H}_5$  ( $2 \leq i \leq 10$ ) using property 4.4<sup>1</sup>.

---

pairwise almost disjoint, i.e.,  $\forall_{A, B \in P} A \neq B \Rightarrow \text{vol}(A \cap B) = 0$ . For the purpose of this proof this is enough, since the integration in sets of measure zero is zero.

<sup>1</sup>We chose to simplify the very sharp enclosures obtained for the rational numbers (with 16 digits precision) with a high order Taylor model, by showing the corresponding fraction.

(a)		(b)	
$B$	$[P_{\mathcal{H}_5}](f, B)$	$B$	$[P_{\mathcal{H}_5}](f, B)$
$B_2$	$[0, 1/4]$	$B_5$	$[5/64]$
$B_3$	$[0, 1/2]$	$B_6$	$[1/64]$
$B_4$	$[1/8]$	$B_7$	$[0, 1/32]$
		$B_8$	$[0, 1/16]$
		$B_9$	$[0, 3/32]$
		$B_{10}$	$[0, 1/8]$

**Table 4.2:** Enclosures for the probability of  $B_i \cap \mathcal{H}_5$ .

Using the values of table 4.2 (a) and property 4.5 we obtain  $P(\mathcal{H}_5) \in [P](\mathcal{H}_{\boxplus_1}, f) = [\frac{1}{8}, \frac{7}{8}]$  which, as expected, encloses the correct value ( $\frac{1}{3}$ ) obtained analytically.

The previous example shows that a specific joint box cover may not provide a tight enclosure for the probability of an event. To get better bounds for the exact probability it is necessary to consider larger inner approximations and/or smaller outer approximations.

**Property 4.6 (Tighter Probability Enclosure)** *Given a PC event  $\mathcal{H}$  and two joint box covers of  $\mathcal{H}$ ,  $\mathcal{H}_{\boxplus_1}$  and  $\mathcal{H}_{\boxplus_2}$ , such that  $\mathcal{H}_{\boxplus_2} \preceq \mathcal{H}_{\boxplus_1}$  then:*

$$wid([P](\mathcal{H}_{\boxplus_2}, f)) \leq wid([P](\mathcal{H}_{\boxplus_1}, f))$$

**Proof.** In the following consider  $[T_0](f, B) = [0, sup([f](B))vol(B)]$  for a box  $B$  and  $\mathcal{H}_{\boxplus} = \mathcal{H}_{\square} \setminus \mathcal{H}_{\blacksquare}$ . For a box  $B_1 \in \mathcal{H}_{\square_1}$  let  $S_{\blacksquare_{B_1}} = \{B_2 \in \mathcal{H}_{\blacksquare_2} : B_2 \subseteq B_1\}$ ,  $S_{\square_{B_1}} = \{B_2 \in \mathcal{H}_{\square_2} : B_2 \subseteq B_1\}$  and  $S_{\boxplus_{B_1}} = \{B_2 \in \mathcal{H}_{\boxplus_2} : B_2 \subseteq B_1\}$ .

By property 4.4,

$$wid([P](\mathcal{H}_{\boxplus_1}, f)) = wid\left(\sum_{B_1 \in \mathcal{H}_{\blacksquare_1}} [I](f, B_1)\right) + wid\left(\sum_{B_1 \in \mathcal{H}_{\square_1}} [T_0](f, B_1)\right)$$

and by property 2.33,

$$\begin{aligned}
 \text{wid}([P](\mathcal{H}_{\boxplus_2}, f)) &= \text{wid} \left( \sum_{B_1 \in \mathcal{H}_{\blacksquare_1}} \sum_{B_2 \in S_{\blacksquare_{B_1}}} [I](f, B_2) \right) + \\
 &\quad \text{wid} \left( \sum_{B_1 \in \mathcal{H}_{\boxminus_1}} \left[ \sum_{B_2 \in S_{\blacksquare_{B_1}}} [I](f, B_2) + \sum_{B_2 \in S_{\boxminus_{B_1}}} [T_0](f, B_2) \right] \right) \quad (4.6)
 \end{aligned}$$

For the first term of the previous sum we have:

$$\begin{aligned}
 \text{wid} \left( \sum_{B_1 \in \mathcal{H}_{\blacksquare_1}} \sum_{B_2 \in S_{\blacksquare_{B_1}}} [I](f, B_2) \right) &= \sum_{B_1 \in \mathcal{H}_{\blacksquare_1}} \text{wid} \left( \sum_{B_2 \in S_{\blacksquare_{B_1}}} [I](f, B_2) \right) \\
 &\leq \sum_{B_1 \in \mathcal{H}_{\blacksquare_1}} \text{wid}([I](f, B_1)) \quad \text{property 3.17} \\
 &= \text{wid} \left( \sum_{B_1 \in \mathcal{H}_{\blacksquare_1}} [I](f, B_1) \right)
 \end{aligned}$$

For the second term, (4.6), of the previous sum we have:

$$\begin{aligned}
 (4.6) &= \sum_{B_1 \in \mathcal{H}_{\boxminus_1}} \left[ \sum_{B_2 \in S_{\blacksquare_{B_1}}} \text{wid}([I](f, B_2)) + \sum_{B_2 \in S_{\boxminus_{B_1}}} \text{wid}([T_0](f, B_2)) \right] \\
 &\leq \sum_{B_1 \in \mathcal{H}_{\boxminus_1}} \left[ \sum_{B_2 \in S_{\blacksquare_{B_1}}} \text{wid}([T_0](f, B_2)) + \sum_{B_2 \in S_{\boxminus_{B_1}}} \text{wid}([T_0](f, B_2)) \right] \\
 &= \sum_{B_1 \in \mathcal{H}_{\boxminus_1}} \sum_{B_2 \in S_{\boxminus_{B_1}}} \text{wid}([T_0](f, B_2)) \\
 &= \sum_{B_1 \in \mathcal{H}_{\boxminus_1}} \sum_{B_2 \in S_{\boxminus_{B_1}}} \text{sup}([f](B_2)) \text{vol}(B_2) \\
 &\leq \sum_{B_1 \in \mathcal{H}_{\boxminus_1}} \sum_{B_2 \in S_{\boxminus_{B_1}}} \text{sup}([f](B_1)) \text{vol}(B_2) \quad \text{monotonicity of } [f] \\
 &= \sum_{B_1 \in \mathcal{H}_{\boxminus_1}} \text{sup}([f](B_1)) \sum_{B_2 \in S_{\boxminus_{B_1}}} \text{vol}(B_2)
 \end{aligned}$$

$$\leq \sum_{B_1 \in \mathcal{H}_{\boxplus_1}} \sup([f](B_1)) \text{vol}(B_1) = \text{wid} \left( \sum_{B_1 \in \mathcal{H}_{\boxplus_1}} [T_0](f, B_1) \right)$$

So,

$$\begin{aligned} \text{wid}([P](\mathcal{H}_{\boxplus_2}, f)) &\leq \text{wid} \left( \sum_{B_1 \in \mathcal{H}_{\blacksquare_1}} [I](f, B_1) \right) + \text{wid} \left( \sum_{B_1 \in \mathcal{H}_{\boxplus_1}} [T_0](f, B_1) \right) \\ &= \text{wid}([P](\mathcal{H}_{\boxplus_1}, f)) \end{aligned}$$

■

The illustration of these concepts in our working example follows.

**Example 4.6.** Consider PC *Triangle* data (from example 4.3), the constraint event  $\mathcal{H}_5$  and its joint box cover  $\mathcal{H}_{\boxplus_2} = \langle \{B_i : i \in \{4, \dots, 10\}\}, \{B_4, B_5, B_6\} \rangle$ . Using the values of table 4.2 and property 4.5 we obtain  $P(\mathcal{H}_5) \in [P](\mathcal{H}_{\boxplus_2}, f) = [\frac{7}{32}, \frac{17}{32}]$ .

Property 4.6 holds for the joint box covers  $\mathcal{H}_{\boxplus_1}$  (example 4.5) and  $\mathcal{H}_{\boxplus_2}$  of  $\mathcal{H}_5$ , where  $\mathcal{H}_{\boxplus_2} \preceq \mathcal{H}_{\boxplus_1}$ , i.e.,  $\text{wid}([\frac{7}{32}, \frac{17}{32}]) = \frac{5}{16} \leq \frac{12}{16} = \text{wid}([\frac{1}{8}, \frac{7}{8}])$ .

## 4.5 Conditional Probability Enclosure

Probabilistic reasoning by conditioning is an important aspect of probability theory as presented in §3.2. This section shows how the proposed PC framework can incorporate this kind of reasoning.

As defined in 3.9, the conditional probability of a PC event  $\mathcal{H}_1$  given another PC event  $\mathcal{H}_2$  is formalized as follows.

**Definition 4.8 (Conditional Probability of a PC Event)** *Given two PC events  $\mathcal{H}_1, \mathcal{H}_2$  where  $P(\mathcal{H}_2) > 0$ , the probability of  $\mathcal{H}_1$  given  $\mathcal{H}_2$  is defined as:*

$$P(\mathcal{H}_1 | \mathcal{H}_2) = \frac{P(\mathcal{H}_1 \cap \mathcal{H}_2)}{P(\mathcal{H}_2)} = \frac{\int_{\mathcal{H}_1 \cap \mathcal{H}_2} f(\mathbf{x}) d\mathbf{x}}{\int_{\mathcal{H}_2} f(\mathbf{x}) d\mathbf{x}}$$

This definition is illustrated in the following example.

**Example 4.7.** Consider the PC and events of example 4.3. The probability of each box event conditioned by the constraint event, i.e.,  $P(\mathcal{H}_i|\mathcal{H}_5)$ , with  $1 \leq i \leq 4$  is given by  $P(\mathcal{H}_i|\mathcal{H}_5) = \frac{P(\mathcal{H}_i \cap \mathcal{H}_5)}{P(\mathcal{H}_5)}$ . So, using the values of table 4.1, we obtain

$$\begin{aligned} P(\mathcal{H}_1|\mathcal{H}_5) &= \frac{P(\mathcal{H}_1 \cap \mathcal{H}_5)}{P(\mathcal{H}_5)} = \frac{0}{1/3} = 0 & P(\mathcal{H}_2|\mathcal{H}_5) &= \frac{P(\mathcal{H}_2 \cap \mathcal{H}_5)}{P(\mathcal{H}_5)} = \frac{1/24}{1/3} = \frac{1}{8} \\ P(\mathcal{H}_3|\mathcal{H}_5) &= \frac{P(\mathcal{H}_3 \cap \mathcal{H}_5)}{P(\mathcal{H}_5)} = \frac{1/6}{1/3} = \frac{1}{2} & P(\mathcal{H}_4|\mathcal{H}_5) &= \frac{P(\mathcal{H}_4 \cap \mathcal{H}_5)}{P(\mathcal{H}_5)} = \frac{1/8}{1/3} = \frac{3}{8} \end{aligned}$$


---

Since the conjunction of events is also an event, and taking into account previous results, the computation of the conditional probability of an event  $\mathcal{H}_1$  given another event  $\mathcal{H}_2$  can be based on an outer box cover of the conditioning event  $\mathcal{H}_2$  (which is also an outer box cover of the event  $\mathcal{H}_1 \cap \mathcal{H}_2$ ).

**Property 4.7 (Conditional Probability through an Outer Box Cover)**

Consider two PC events  $\mathcal{H}_1, \mathcal{H}_2$  where  $P(\mathcal{H}_2) > 0$  and an outer box cover  $\mathcal{H}_{\square_2}$  of  $\mathcal{H}_2$  (and of  $\mathcal{H}_{\cap} = \mathcal{H}_1 \cap \mathcal{H}_2$ ). The conditional probability of  $\mathcal{H}_1$  given  $\mathcal{H}_2$  is:

$$P(\mathcal{H}_1|\mathcal{H}_2) = \frac{\sum_{B \in \mathcal{H}_{\square_2}} P_{\mathcal{H}_{\cap}}(B)}{\sum_{B \in \mathcal{H}_{\square_2}} P_{\mathcal{H}_2}(B)}$$

**Proof.** Since  $\mathcal{H}_{\square_2}$  is an outer box cover of  $\mathcal{H}_2$  and of  $\mathcal{H}_{\cap}$ , from property 4.1 we have:

$$P(\mathcal{H}_{\cap}) = \sum_{B \in \mathcal{H}_{\square_2}} P_{\mathcal{H}_{\cap}}(B) \quad \text{and} \quad P(\mathcal{H}_2) = \sum_{B \in \mathcal{H}_{\square_2}} P_{\mathcal{H}_2}(B)$$

So, from definition 4.8, property 4.7 holds. ■

**Example 4.8.** Consider the PC and events of example 4.3 and the boxes of example 4.4, where  $\mathcal{H}_{\square_5} = \{B_j : 4 \leq j \leq 10\}$  is an outer box cover of  $\mathcal{H}_5$ . By property 4.7 and using the values of table 4.1:

$$P(\mathcal{H}_1 \cap \mathcal{H}_5) = \sum_{j=4}^{10} P(B_j \cap \mathcal{H}_1 \cap \mathcal{H}_5) = \sum_{j=4}^{10} P(\emptyset) = 0$$

$$P(\mathcal{H}_2 \cap \mathcal{H}_5) = \sum_{j=4}^{10} P(B_j \cap \mathcal{H}_2 \cap \mathcal{H}_5) = P(B_6) + P(B_7 \cap \mathcal{H}_5) + P(B_8 \cap \mathcal{H}_5) = \frac{1}{24}$$

$$P(\mathcal{H}_3 \cap \mathcal{H}_5) = \sum_{j=4}^{10} P(B_j \cap \mathcal{H}_3 \cap \mathcal{H}_5) = P(B_5) + P(B_9 \cap \mathcal{H}_5) + P(B_{10} \cap \mathcal{H}_5) = \frac{1}{6}$$

$$P(\mathcal{H}_4 \cap \mathcal{H}_5) = \sum_{j=4}^{10} P(B_j \cap \mathcal{H}_4 \cap \mathcal{H}_5) = P(\mathcal{H}_4) = \frac{1}{8}$$

Consequently,

$$\begin{aligned} P(\mathcal{H}_1|\mathcal{H}_5) &= \frac{0}{1/3} = 0 & P(\mathcal{H}_2|\mathcal{H}_5) &= \frac{1/24}{1/3} = \frac{1}{8} \\ P(\mathcal{H}_3|\mathcal{H}_5) &= \frac{1/8}{1/3} = \frac{1}{2} & P(\mathcal{H}_4|\mathcal{H}_5) &= \frac{1/6}{1/3} = \frac{3}{8} \end{aligned}$$

As expected, the results obtained using the outer box cover are exactly the same as in the previous example.

In property 4.7 the same outer box cover is used for both  $\mathcal{H}_2$  and  $\mathcal{H}_\cap$ . However, since  $\mathcal{H}_\cap \subseteq \mathcal{H}_2$ , there might be tighter outer box covers for it which allow the computation of narrower enclosures for conditional probabilities.

**Property 4.8 (Enclosure for the Conditional Probability)** *Consider two events  $\mathcal{H}_1$  and  $\mathcal{H}_2$ , where  $P(\mathcal{H}_2) > 0$ , a joint box cover  $\mathcal{H}_{\boxplus_2}$  of  $\mathcal{H}_2$  and a joint box cover  $\mathcal{H}_{\boxplus_\cap}$  of  $\mathcal{H}_\cap = \mathcal{H}_1 \cap \mathcal{H}_2$ . An enclosure for the conditional probability of  $\mathcal{H}_1$  given  $\mathcal{H}_2$  is given by:*

$$P(\mathcal{H}_1|\mathcal{H}_2) \in [P](\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2}, f) = \frac{\sum_{B \in \mathcal{H}_{\boxplus_\cap}} [P_{\mathcal{H}_{\boxplus_\cap}}](f, B)}{\sum_{B \in \mathcal{H}_{\boxplus_2}} [P_{\mathcal{H}_{\boxplus_2}}](f, B)}$$

**Proof.** Since  $P(\mathcal{H}_1|\mathcal{H}_2) = \frac{P(\mathcal{H}_1 \cap \mathcal{H}_2)}{P(\mathcal{H}_2)}$  (see definition 4.8) and, from property 4.5

$$\begin{aligned} P(\mathcal{H}_1 \cap \mathcal{H}_2) \in [P](\mathcal{H}_{\boxplus_1}, f) &= \sum_{B \in \mathcal{H}_{\boxplus_1}} [P_{\mathcal{H}_1}](f, B) \\ P(\mathcal{H}_2) \in [P](\mathcal{H}_{\boxplus_2}, f) &= \sum_{B \in \mathcal{H}_{\boxplus_2}} [P_{\mathcal{H}_2}](f, B) \end{aligned}$$

then, by the properties of interval arithmetic,

$$P(\mathcal{H}_1|\mathcal{H}_2) \in [P](\mathcal{H}_{\boxplus_1}, \mathcal{H}_{\boxplus_2}, f) = \frac{[P](\mathcal{H}_{\boxplus_1}, f)}{[P](\mathcal{H}_{\boxplus_2}, f)}$$

■

**Example 4.9.** Consider the PC *Triangle* (see example 4.3), the event  $\mathcal{H}_5$  with  $[P](\mathcal{H}_{\boxplus_2}, f) = [\frac{7}{32}, \frac{17}{32}]$  ( $\mathcal{H}_{\boxplus_2}$  is a joint box cover of  $\mathcal{H}_5$ , see example 4.6), the event  $\mathcal{H}_\cap = \mathcal{H}_2 \cap \mathcal{H}_5$  (with  $P(\mathcal{H}_2 \cap \mathcal{H}_5) = \frac{1}{24}$ , see table 4.1) and its joint box cover  $\mathcal{H}_{\boxplus_\cap} = \langle \{B_6, B_7, B_8\}, \{B_6\} \rangle$  (see figure 4.3(a)). Using the values of table 4.2 and property 4.5 we obtain:

$$P(\mathcal{H}_2 \cap \mathcal{H}_5) \in \sum_{i=6}^8 [P_{\mathcal{H}_\cap}](f, B_i) = \left[ \frac{1}{64}, \frac{7}{64} \right]$$

Then, by property 4.8 we obtain:

$$P(\mathcal{H}_2|\mathcal{H}_5) = \frac{1}{8} \in \frac{[P](\mathcal{H}_{\boxplus_\cap}, f)}{[P](\mathcal{H}_{\boxplus_2}, f)} = \frac{[\frac{1}{64}, \frac{7}{64}]}{[\frac{7}{32}, \frac{17}{32}]} = \left[ \frac{1}{34}, \frac{1}{2} \right]$$

## 4.6 Algorithms

This section presents algorithms to compute probability and conditional probability enclosures of PC events as presented in the previous sections.

The generic function *cReasoning* (algorithm 3, page 40), which computes joint box covers for the feasible space of a CCSP, is the core of several algorithms proposed in

this and the next chapter. It is parameterizable by using distinct *inner*, *eligible* and *stop* predicates, *split* functions and *order* criteria (see section A.1 in appendix A). The default parametrization is presented and, for simplicity, the corresponding algorithm is denoted as  $cReasoning_{prob}$  with the default parameters omitted.

**Property 4.9 (Default parametrization for  $cReasoning$ )** *The default parametrization for function  $cReasoning(\mathcal{F}_{\boxplus}, C, inner, split, eligible, order, stop)$  (algorithm 3) considers defaults of section A.1 (appendix A) and parameterizes:*

- *the stop predicate as  $stop_{\delta}(\mathcal{H}_{\boxplus}) \equiv wid([P](\mathcal{H})) \leq \delta$ , imposing a specified accuracy  $\delta$  to the probability enclosure computed over its joint box cover argument (as presented in property 4.5).*
- *the order criteria as  $order_P$ , imposing a choice of boxes by decreasing order of uncertainty in their probability, i.e. in decreasing order of  $wid([P](B))$ . So, boxes with largest uncertainty in their probability are chosen for processing, contributing to larger decreases of the uncertainty of the total probability.*

#### 4.6.1 Probability Enclosure

Algorithm 6 computes an enclosure for the probability of a PC event  $P(\mathcal{H})$  according to property 4.5.

**Algorithm 6:**  $probabilityEnclosure(\langle\langle X, D, C_{\mathcal{H}} \rangle, f \rangle, \varepsilon, \delta)$

**Input:**  $\langle\langle X, D, C_{\mathcal{H}} \rangle, f \rangle$ : PC;  $\varepsilon, \delta$ : double  
**Output:**  $[P](\mathcal{H})$ : interval;  
1  $\langle\mathcal{H}_{\square}, \mathcal{H}_{\blacksquare}\rangle \leftarrow cReasoning_{prob}(\langle\{D\}, \emptyset, C_{\mathcal{H}}\rangle$ ;  
2 **return**  $[P](\langle\mathcal{H}_{\square}, \mathcal{H}_{\blacksquare}\rangle, f)$ ;

Considering a PC where its set of constraints,  $C_{\mathcal{H}}$ , is associated with the PC event  $\mathcal{H}$ , algorithm 6 yields increasingly tighter covers by successive applications of  $crStep$  (algorithm 2, page 39), within  $cReasoning_{prob}$ , until the intended precision  $\delta$  for the probability is reached (line 1). The final outer box cover for  $\mathcal{H}$  is used to return the interval that encloses the exact probability of  $\mathcal{H}$  (line 2). The parametrization of the  $cReasoning_{prob}$  algorithm (see property 4.9) directs it to reduce the uncertainty on the

$C_1$	$x^2y + y^2x \leq 0.5$	$C_{\mathcal{H}_1} = \{C_1\}$
$C_2$	$\cos(3x) + \cos(3y) \leq 0.5$	$C_{\mathcal{H}_2} = \{C_2\}$

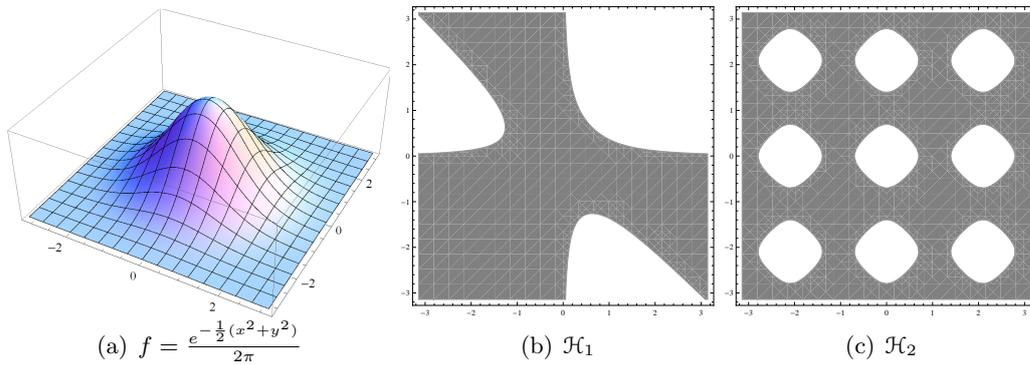
**Table 4.3:** Definition of constraints and events.

total probability, by choosing the boxes with higher uncertainty in their probability and thus successively approaching the stopping condition.

Algorithm 6 is correct and converges to the exact probability value (see property 4.10). Its correctness is guaranteed because the initial pair  $\langle \{D\}, \emptyset \rangle$  is a joint box cover of  $\mathcal{H} = \mathcal{F}(\langle X, D, C_{\mathcal{H}} \rangle)$  and, inside  $cReasoning_{prob}$ , every iteration of  $crStep$  produces a tighter joint box cover of  $\mathcal{H}$ . So, when  $cReasoning_{prob}$  stops  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle$  is a joint box cover of  $\mathcal{H}$  and, by property 4.5,  $P(\mathcal{H}) \in [P](\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle, f)$ .

The algorithm is an abstraction of the real procedure that is implemented. In practice the algorithms that rely on the probability enclosure as a stopping condition maintain it during the algorithm and update such enclosure as the joint box cover is updated. In this particular case, that enclosure is the return value and is not recomputed in the end.

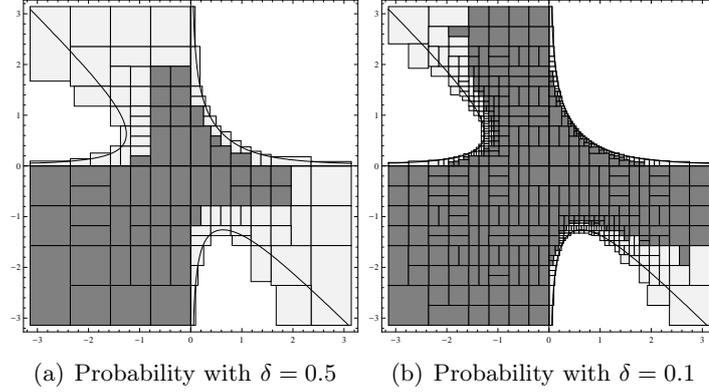
**Example 4.10.** Consider PC  $\langle \langle X, D, C \rangle, f \rangle$  where  $X = \langle x, y \rangle$ ,  $D = [-\pi, \pi] \times [-\pi, \pi]$ ,  $C = \{C_1, C_2\}$  (defined in Table 4.3) and  $f = \frac{e^{-\frac{1}{2}(x^2+y^2)}}{2\pi}$  is the bi-normal distribution (figure 4.4 (a)) defined over  $\Omega = \mathbb{R}^2$ .  $\mathcal{H}_1$  and  $\mathcal{H}_2$  are PC events (shown in figure 4.4).



**Figure 4.4:** Events and PDF used in examples.

Figure 4.5 presents the covers resulting from applying algorithm 6 to compute enclosures for the probability of the PC event  $\mathcal{H}_1$  with  $\delta = 0.5$  and  $\delta = 0.1$ . Notice that the *inner*<sub>d</sub>

predicate was not able to identify, as inner boxes, some boxes completely included in the feasible space.



**Figure 4.5:** Joint box covers of the PC event  $\mathcal{H}_1$  computed when using algorithm 6 with a safe integration method. The boundary boxes are light gray and the inner boxes are gray.

To study the convergence of algorithm 6 we assume that the algorithm is implemented with an infinite precision interval arithmetic.

**Property 4.10 (Algorithm Convergence)** *Let  $\langle\langle X, D, C \rangle, f \rangle$  be a PC and  $\mathcal{H} = \mathcal{F}(\langle X, D_{\mathcal{H}}, C_{\mathcal{H}} \rangle)$  a PC event, with  $D_{\mathcal{H}} \subseteq D$  and  $C_{\mathcal{H}} \subseteq C$ .*

*Consider a sequence  $(\mathcal{H}_{\boxplus_k})_{k \in \mathbb{N}}$  computed by  $\text{probabilityEnclosure}(\langle X, D_{\mathcal{H}}, C_{\mathcal{H}} \rangle, 0, 0)$  such that  $\mathcal{H}_{\boxplus_k} = \text{crStep}(\mathcal{H}_{\boxplus_{k-1}}, C_{\mathcal{H}}, \text{split}_2, \text{inner}_d, \text{eligible}_\varepsilon, \text{order}_P)$ . Then*

$$\lim_{k \rightarrow \infty} \text{wid}([P](\mathcal{H}_{\boxplus_k}, f)) = 0.$$

**Proof.** This proof is based on a similar one given in [48].

In the following consider  $\mathcal{H}'_{\blacksquare} = \{B \in \mathcal{H}_{\blacksquare} : \text{wid}([P_{\mathcal{H}}](f, B)) > 0\}$ , sets of boundary boxes  $\mathcal{H}_{\square} \setminus \mathcal{H}_{\blacksquare}$  are represented by  $\mathcal{H}_{\square}$  and

$$\varepsilon_k = \max_{B \in \mathcal{H}'_{\blacksquare_k}} \text{wid}(B). \quad (4.7)$$

Since the conjunction of the *orderP* criterion and *eligible*<sub>0</sub> predicate defines a fair selection strategy wrt boxes in  $\mathcal{H}'_{\blacksquare_k}$  and *split*<sub>2</sub> is fair, then when  $k$  approaches infinity the width of all boxes in  $\mathcal{H}'_{\blacksquare_k}$  approaches zero and so does  $\varepsilon_k$ .

For  $\epsilon > 0$ ,  $\bar{f}_\epsilon$  is the  $\epsilon$ -maximum width of the interval resulting from evaluating  $[f]$  over every box smaller than  $\epsilon$ :

$$\bar{f}_\epsilon = \max\{\text{wid}([f](B)) : B \subseteq D \wedge \text{wid}(B) \leq \epsilon\}. \quad (4.8)$$

Since  $[f]$  is a convergent inclusion function,  $\lim_{\epsilon \rightarrow 0} \bar{f}_\epsilon = 0$ .

Algorithm *cReasoning<sub>prob</sub>* with  $\delta = 0$  and  $\varepsilon = 0$  belongs to the *cReasoning<sub>∞</sub>* family of algorithms since *split*<sub>2</sub> is fair and the conjunction of the *orderP* criterion and *eligible*<sub>0</sub> predicate define a fair selection strategy wrt boundary boxes<sup>1</sup> (see property 2.5).

We have

$$\begin{aligned} [P](\mathcal{H}_{\boxplus_k}, f) &= \sum_{B \in \mathcal{H}_{\square_k}} [P_{\mathcal{H}c}](f, B) \\ &= \sum_{B \in \mathcal{H}'_{\blacksquare_k}} [I](f, B) + \sum_{B \in \mathcal{H}_{\square_k}} [0, \text{sup}([f](B))\text{vol}(B)] \end{aligned}$$

and so

$$\text{wid}([P](\mathcal{H}_{\boxplus_k}, f)) = \sum_{B \in \mathcal{H}'_{\blacksquare_k}} \text{wid}([I](f, B)) + \sum_{B \in \mathcal{H}_{\square_k}} \text{wid}([0, \text{sup}([f](B))\text{vol}(B)])$$

We now prove that both terms of the above sum converge to zero as  $k$  approaches infinity.

For the first term we have:

$$\begin{aligned} \sum_{B \in \mathcal{H}'_{\blacksquare_k}} \text{wid}([I](f, B)) &= \sum_{B \in \mathcal{H}'_{\blacksquare_k}} \text{wid}([I](f, B)) \\ &= \sum_{B \in \mathcal{H}'_{\blacksquare_k}} \text{wid}([I]^*(f, B) \cap [f](B)\text{vol}(B)) \quad \text{definition 3.36} \end{aligned}$$

---

<sup>1</sup>Notice that *orderP* will not choose boxes with  $\text{wid}([P_{\mathcal{H}c}](f, B)) = 0$ , but there are no boundary boxes on such situation.

$$\begin{aligned}
&\leq \sum_{B \in \mathcal{H}'_{\blacksquare_k}} \text{wid}([f](B)) \text{vol}(B) \\
&\leq \sum_{B \in \mathcal{H}'_{\blacksquare_k}} \bar{f}_{\varepsilon_k} \text{vol}(B) && \text{formulas (4.7) and (4.8)} \\
&= \bar{f}_{\varepsilon_k} \sum_{B \in \mathcal{H}'_{\blacksquare_k}} \text{vol}(B) \\
&\leq \bar{f}_{\varepsilon_k} \text{vol}(D) && \bigcup \mathcal{H}'_{\blacksquare_k} \subseteq D
\end{aligned}$$

Since  $\bar{f}_{\varepsilon_k}$  converges to zero as  $k$  approaches infinity and given that  $\text{vol}(D)$  is a constant, the whole term converges to zero as  $k$  approaches infinity.

For the second term we have:

$$\begin{aligned}
\sum_{B \in \mathcal{H}_{\square_k}} \text{wid}([0, \text{sup}([f](B))\text{vol}(B)]) &= \sum_{B \in \mathcal{H}_{\square_k}} \text{sup}([f](B))\text{vol}(B) \\
&\leq \sum_{B \in \mathcal{H}_{\square_k}} \text{sup}([f](D))\text{vol}(B) && \text{monotonicity of } [f] \\
&= \text{sup}([f](D)) \sum_{B \in \mathcal{H}_{\square_k}} \text{vol}(B)
\end{aligned}$$

Since  $\sum_{B \in \mathcal{H}_{\square_k}} \text{vol}(B) = \text{vol}(\Delta_k \mathcal{H})$ , by property 2.5, it converges to zero as  $k$  approaches infinity. Given that  $\text{sup}([f](D))$  is a constant, the whole term converges to zero as  $k$  approaches infinity.  $\blacksquare$

In practice, computational limitations imposed by the floating-point system and interval arithmetic, make it impossible for algorithm 6 to tighten the probability bounds beyond a certain threshold. In fact, the algorithm may stop before reaching the desired precision if  $\varepsilon$  forces *cReasoning<sub>prob</sub>* algorithm to end before the uncertainty on the probability enclosure is less than  $\delta$ .

### 4.6.2 Conditional Probability Enclosure

Property 4.8 uses two distinct joint box covers: one for the conditioning event  $\mathcal{H}_2$  (the feasible space of  $\langle X, D, C_{\mathcal{H}_2} \rangle$ ) and another for the conjunction of events  $\mathcal{H}_1 \cap \mathcal{H}_2$  (the feasible space of  $\langle X, D, C_{\mathcal{H}_1} \cup C_{\mathcal{H}_2} \rangle$ ). So, we can use algorithm 6 to compute, independently, the probabilities of both the conditioning event and the conjunction of

events, maintaining two distinct joint box covers. Nevertheless these two covers have common boxes and a more efficient algorithm, that avoids duplicate box processing, can be exploited.

This algorithm (algorithm 7) relies on a dedicated constraint reasoning algorithm (algorithm 8) that computes simultaneously the two joint box covers. Considering a PC, two sets of constraints,  $C_{\mathcal{H}_1}$  and  $C_{\mathcal{H}_2}$ , associated with, respectively, the conditioned event and the conditioning event,  $cReasConj$  yields increasingly tighter joint box covers of both  $\mathcal{H}_\cap$  and  $\mathcal{H}_2$ , until the required accuracy  $\delta$  for the conditional probability is reached (line 2). The final joint box covers are used to compute an enclosure for the exact conditional probability of  $\mathcal{H}_1$  given  $\mathcal{H}_2$  (line 3). The function parameters for  $cReasConj$  are similar to those of algorithm 6.

**Algorithm 7:**  $conditionalProbabilityEnclosure(\langle\langle X, D, C \rangle, f \rangle, C_{\mathcal{H}_1}, C_{\mathcal{H}_2}, \varepsilon, \delta)$

<p><b>Input:</b> <math>\langle\langle X, D, C \rangle, f \rangle</math>: PC; <math>C_{\mathcal{H}_1}, C_{\mathcal{H}_2}</math>: set of constraints; <math>\varepsilon, \delta</math>: double  <b>Output:</b> <math>I</math>: interval;</p> <ol style="list-style-type: none"> <li>1 <math>stop_\delta(\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2}) \equiv wid([P](\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2}, f)) \leq \delta</math>;</li> <li>2 <math>\mathcal{H}_{\boxplus} \leftarrow \langle\{D\}, \emptyset\rangle</math>;</li> <li>3 <math>\langle\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2}\rangle \leftarrow</math>  <math>cReasConj(\mathcal{H}_{\boxplus}, \mathcal{H}_{\boxplus}, C_{\mathcal{H}_1}, C_{\mathcal{H}_2}, split_2, inner_d, eligible_\varepsilon, order_P, stop_\delta)</math>;</li> <li>4 <b>return</b> <math>[P](\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2}, f)</math>;</li> </ol>
--

The correctness of algorithm 7 is guaranteed because the initial pair  $\langle\{D\}, \emptyset\rangle$  is a joint box cover of both  $\mathcal{H}_2 = \mathcal{F}(\langle X, D, C_{\mathcal{H}_2} \rangle)$  and  $\mathcal{H}_\cap = \mathcal{F}(\langle X, D, C_{\mathcal{H}_1} \cup C_{\mathcal{H}_2} \rangle)$  and, inside  $cReasConj$ , every iteration of  $crStep$  produces tighter joint box covers of  $\mathcal{H}_2$  and  $\mathcal{H}_\cap$ . So, when  $cReasConj$  stops  $\mathcal{H}_{\boxplus_2}$  and  $\mathcal{H}_{\boxplus_\cap}$ , are a joint box cover of, respectively,  $\mathcal{H}_2$  and  $\mathcal{H}_\cap$ . Then, by property 4.8,  $P(\mathcal{H}_1|\mathcal{H}_2) \in [P](\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2}, f)$ .

Algorithm 8 is an adaptation of algorithm 3, but receives an additional set of constraints  $C_{\mathcal{H}_2}$  and maintains two joint box covers. It successively updates its joint box covers (lines 1 – 16) until the stopping criterion is reached (line 1), returning them in the end (line 17).

The boxes to process are chosen among the boxes of the two covers (line 2). Unless the chosen box  $B$  is a common boundary box, the joint box covers are updated as in algorithm 3, with the boxes resulting from the split. When it is a common boundary box

**Algorithm 8:**  $cReasConj(\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2}, C_{\mathcal{H}_1}, C_{\mathcal{H}_2}, split, inner, eligible, order, stop)$

**Input:**  $\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2}$ : joint box covers;  $C_{\mathcal{H}_1}, C_{\mathcal{H}_2}$ : set of constraints;  $split, order$ : function;  $inner, eligible, stop$ : predicate;

**Output:**  $\langle \mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2} \rangle$ : pair of joint box covers

```

1 while ( $\neg stop(\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2})$ ) do
2   if ( $B \leftarrow choose(\mathcal{H}_{\square_2} \cup \mathcal{H}_{\square_\cap}, eligible, order) = \emptyset$ ) then break;
3    $S \leftarrow split(B)$ ;
4   if  $B \in \mathcal{H}_{\blacksquare_2} \vee B \in \mathcal{H}_{\blacksquare_\cap}$  then
5     if  $B \in \mathcal{H}_{\blacksquare_2}$  then  $\langle \mathcal{H}_{\square_2}, \mathcal{H}_{\blacksquare_2} \rangle \leftarrow \langle (\mathcal{H}_{\square_2} \setminus \{B\}) \cup S, (\mathcal{H}_{\blacksquare_2} \setminus \{B\}) \cup S \rangle$ ;
6     if  $B \in \mathcal{H}_{\blacksquare_\cap}$  then  $\langle \mathcal{H}_{\square_\cap}, \mathcal{H}_{\blacksquare_\cap} \rangle \leftarrow \langle (\mathcal{H}_{\square_\cap} \setminus \{B\}) \cup S, (\mathcal{H}_{\blacksquare_\cap} \setminus \{B\}) \cup S \rangle$ ;
7   else if  $B \in \mathcal{H}_{\square_2} \wedge B \in \mathcal{H}_{\square_\cap}$  then
8      $\langle \mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2} \rangle \leftarrow commonBoundaryBoxes(\mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2}, C_{\mathcal{H}_1}, C_{\mathcal{H}_2}, B, S)$ ;
9   else
10    if  $B \in \mathcal{H}_{\square_2}$  then  $C \leftarrow C_{\mathcal{H}_2}$ ;  $\mathcal{H}_{\blacksquare} \leftarrow \mathcal{H}_{\blacksquare_2}$ ;  $\mathcal{H}_{\square} \leftarrow \mathcal{H}_{\square_2}$ ;
11    else  $C \leftarrow C_{\mathcal{H}_1}$ ;  $\mathcal{H}_{\blacksquare} \leftarrow \mathcal{H}_{\blacksquare_\cap}$ ;  $\mathcal{H}_{\square} \leftarrow \mathcal{H}_{\square_\cap}$ ;
12     $L_{\square} \leftarrow \{CPA(B_i, C) : B_i \in S\}$ ;
13     $\mathcal{H}_{\blacksquare} \leftarrow \mathcal{H}_{\blacksquare} \cup \{B_i \in L_{\square} : inner(B_i, C)\}$ ;
14     $\mathcal{H}_{\square} \leftarrow (\mathcal{H}_{\square} \setminus \{B\}) \cup L_{\square}$ ;
15  end
16 end
17 return  $\langle \mathcal{H}_{\boxplus_\cap}, \mathcal{H}_{\boxplus_2} \rangle$ ;

```

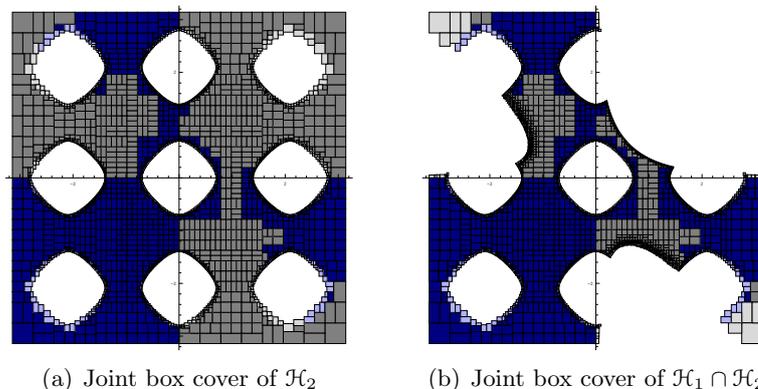
(line 7), the boxes in  $S$  are processed by function *commonBoundaryBoxes* (algorithm 9), which adequately updates both joint box covers.

<p><b>Algorithm 9:</b> <i>commonBoundaryBoxes</i>(<math>\mathcal{H}_{\boxplus_{\cap}}, \mathcal{H}_{\boxplus_2}, C_{\mathcal{H}_1}, C_{\mathcal{H}_2}, B, S</math>)</p> <p><b>Input:</b> <math>\mathcal{H}_{\boxplus_{\cap}}, \mathcal{H}_{\boxplus_2}</math>: joint box covers; <math>C_{\mathcal{H}_1}, C_{\mathcal{H}_2}</math>: set of constraints; <math>B</math>: box; <math>S</math>: set of boxes</p> <p><b>Output:</b> <math>\langle \mathcal{H}_{\boxplus_{\cap}}, \mathcal{H}_{\boxplus_2} \rangle_{out}</math>: pair of joint box covers;</p> <pre> 1  foreach <math>B_i \in S</math> do 2      if <math>(N_2 \leftarrow \mathcal{CPA}(B_i, C_{\mathcal{H}_2})) \neq \emptyset</math> then 3          <math>\mathcal{H}_{\square_2} \leftarrow \mathcal{H}_{\square_2} \cup \{N_2\}</math>;   if <math>inner(N_2, C_{\mathcal{H}_2})</math> then <math>\mathcal{H}_{\blacksquare_2} \leftarrow \mathcal{H}_{\blacksquare_2} \cup \{N_2\}</math>; 4          if <math>(N_{\cap} \leftarrow \mathcal{CPA}(N_2, C_{\mathcal{H}_1})) \neq \emptyset</math> then 5              if <math>inner(N_2, C_{\mathcal{H}_1})</math> then 6                  <math>\mathcal{H}_{\square_{\cap}} \leftarrow \mathcal{H}_{\square_{\cap}} \cup \{N_2\}</math>; 7                  <math>\mathcal{H}_{\blacksquare_{\cap}} \leftarrow \mathcal{H}_{\blacksquare_{\cap}} \cup \{N_2\}</math>; 8              end 9              else if <math>inner(N_2, C_{\mathcal{H}_2})</math> then <math>\mathcal{H}_{\square_{\cap}} \leftarrow \mathcal{H}_{\square_{\cap}} \cup \{N_{\cap}\}</math>; 10             else <math>\mathcal{H}_{\square_{\cap}} \leftarrow \mathcal{H}_{\square_{\cap}} \cup \{N_2\}</math>; 11         end 12     end 13 end 14 return <math>\langle \langle \mathcal{H}_{\square_{\cap}} \setminus \{B\}, \mathcal{H}_{\blacksquare_{\cap}} \rangle, \langle \mathcal{H}_{\square_2} \setminus \{B\}, \mathcal{H}_{\blacksquare_2} \rangle \rangle</math>;                 </pre>
---

Algorithm 9 processes a set of boxes  $S$ , resulting from the split of a common boundary box  $B$ . Whenever a box  $B_i$  from  $S$  results in an inner box of  $\mathcal{H}_2$  (when narrowed wrt  $C_{\mathcal{H}_2} - N_2$ ) but in a boundary box of  $\mathcal{H}_{\cap}$  (when  $N_2$  is narrowed wrt  $C_{\mathcal{H}_1} - N_{\cap}$ ) then  $N_2$  is added to the inner and outer box covers of  $\mathcal{H}_2$  and  $N_{\cap}$  is added to the outer box cover of  $\mathcal{H}_{\cap}$ . From now on each box,  $N_2$  and  $N_{\cap}$ , is processed independently in its corresponding cover. In every other case the processing of each box  $B_i$  from  $S$  is similar to that of algorithm 3, but two covers must be updated.

This procedure avoids duplicated processing of boxes, keeping the same box in both covers whenever it is a common inner or boundary box. Only when a common box has different roles in each joint box cover (inner box for  $\mathcal{H}_2$  and boundary box for  $\mathcal{H}_{\cap}$ ) it is duplicated and the boxes are processed independently thereafter.

**Example 4.11.** Consider the PC and events of example 4.10. Figure 4.6 presents the covers resulting from applying algorithm 7 to compute an enclosure for the conditional probability of  $\mathcal{H}_1$  given  $\mathcal{H}_2$ , with  $\delta = 0.05$ . It shows the joint box cover of (a)  $\mathcal{H}_2$  and (b)  $\mathcal{H}_1 \cap \mathcal{H}_2$ .



**Figure 4.6:** Joint box covers computed when using algorithm 7 with  $\delta = 0.05$ . The common/other boundary boxes are light blue/gray, the common/other inner boxes are dark blue/gray.

## 4.7 Alternative Approximate Computations

When safety is not a major concern, an alternative approach can be used to compute an approximate enclosure for the integral of a function over a region defined by the conjunction of a box and a PC event. This can be done adopting to Monte Carlo integration as presented in section 3.5.3. In fact the approximate value computed as defined in 3.37<sup>1</sup> can be immediately adopted to obtain an estimate for the value of  $\int_{\mathcal{H} \cap B} h(\mathbf{x}) d\mathbf{x}$ . Since there is an error estimate associated with the value computed in definition 3.37 (the standard deviation of the estimate in definition 3.38), we assume an approximate interval that should contain the exact value of the integral of  $h(\mathbf{x})$  over the conjunction of a box and a PC event.

### Definition 4.9 (Approximate Enclosure for the Integral of $h$ over $(B \cap \mathcal{H})$ )

Consider a PC event  $\mathcal{H}$ , a joint box cover  $\langle \mathcal{H}_\square, \mathcal{H}_\blacksquare \rangle$  of  $\mathcal{H}$  and a function  $h : \mathbb{R}^n \rightarrow \mathbb{R}$ . An approximate enclosure for the integral of  $h$  over  $\mathcal{H} \cap B$ , for any box  $B \in \mathcal{H}_\square$ , is:

$$\forall B \in \mathcal{H}_\square \quad [I_{\mathcal{H}}](h, B) \approx [\widehat{I}_{\mathcal{H}}](h, B) = [\widehat{I}_{\mathcal{H}}(h, B) - \sigma(\widehat{I}_{\mathcal{H}}(h, B)), \widehat{I}_{\mathcal{H}}(h, B) + \sigma(\widehat{I}_{\mathcal{H}}(h, B))]$$

<sup>1</sup>The indicator function  $1_{\mathcal{H}}$  is implemented with the *inner* predicate for degenerated box arguments.

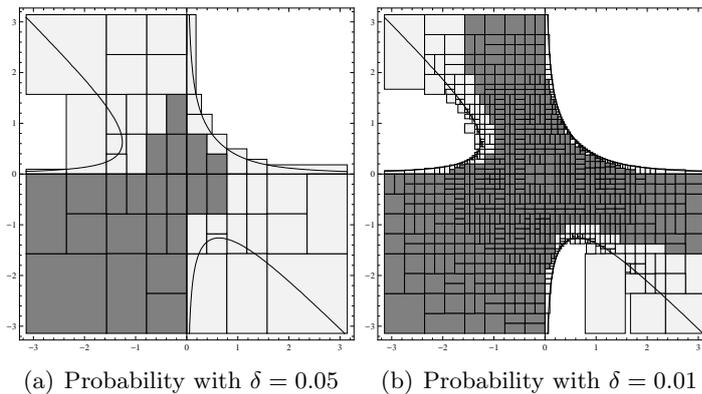
where  $\widehat{I}_{\mathcal{H}}$  is defined 3.37 and  $\sigma(I_{MC}(B))$  is defined in 3.38.

From this generic definition, an approximate enclosure for the probability of the conjunction of a box and a PC event  $\widehat{[P_{\mathcal{H}}]}(f, B)$  is given by  $\widehat{[I_{\mathcal{H}}]}(f, B)$ .

Replacing the safe enclosures  $[P_{\mathcal{H}_i}](f, B)$  by the approximate ones,  $\widehat{[P_{\mathcal{H}_i}]}(B, f)$ , in properties 4.5 and 4.8 will originate approximate enclosures for the probability and conditional probability of an event, respectively  $\widehat{[P]}(\mathcal{H}_{\boxplus}, f)$  and  $\widehat{[P]}(\mathcal{H}_{\boxplus\cap}, \mathcal{H}_{\boxplus 2}, f)$ . Nevertheless, due to the approximate nature of this approach, property 4.6 is no longer verified when safe enclosures are replaced by the corresponding approximate ones.

In algorithms 6 and 7 when safe enclosures (both in the pseudo code and in the *cReasoning* parametrization) are replaced by the corresponding approximate enclosures, there is a loss of robustness and of the convergence properties.

Figure 4.7 presents the covers resulting from applying algorithm 6 with the approximate integration method to compute enclosures for the probability of the PC event  $\mathcal{H}_1$  with  $\delta = 0.05$  and  $\delta = 0.01$ .



**Figure 4.7:** Joint box covers of the PC event  $\mathcal{H}_1$  computed when using algorithm 6 with an approximate integration method. The boundary boxes are light gray and the inner boxes are gray.

## 4.8 Experimental Results

In this section we present the results of applying the algorithms proposed in the previous section to a set of events. Some results are compared with those obtained with *Mathematica* v8.0.1.0 [119]. All the experiments were performed on an Intel Core Duo at 1.83 GHz with 1 GB of RAM.

We consider two versions of the algorithms: PCTM uses the validated quadrature method based on Taylor models (in these experiments we use Taylor Models of order 2) and provides safe enclosures of the desired quantities; PCMC uses Monte Carlo integration (with  $N = 100$  random sample points) and provides estimates for the computed quantities.

### 4.8.1 Development Environment

The proposed algorithms in this thesis were implemented over the interval-based solver RealPaver 1.0 [52], using the C++ programming language and following an object-oriented design. The result is an operational prototype application that can readily be used to test new problems.

RealPaver is an open source, continuous constraint programming framework whose constraint solving engine implements a branch-and-prune algorithm that can be articulated with state-of-the-art constraint propagation and consistency techniques. By default, RealPaver uses *BC4* [14] (see section 2.3.2) to efficiently combine hull and box consistency. Interval arithmetic computations are supported by Gaol, an open source C++ library for interval arithmetic [1] that guarantees correct rounding.

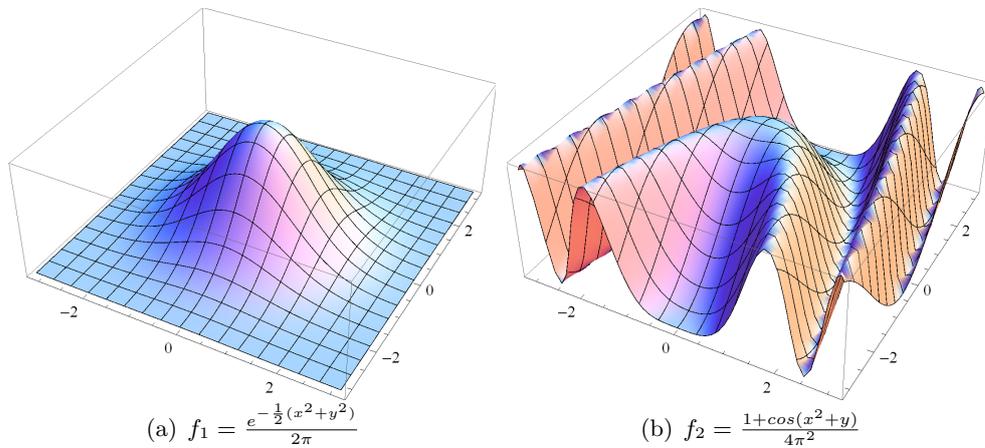
RealPaver also provides a modeling language to easily model constraint systems. The modular design of RealPaver [51] makes it easy to extend with new box splitting methods, choice strategies for bisecting domains and search strategies (e.g. worst-first search strategy). Therefore it is ideal to implement *crStep* and *cReasoning* (chapter 2, pages 39 and 40) algorithms which require customization (different *values* for their functional parameters).

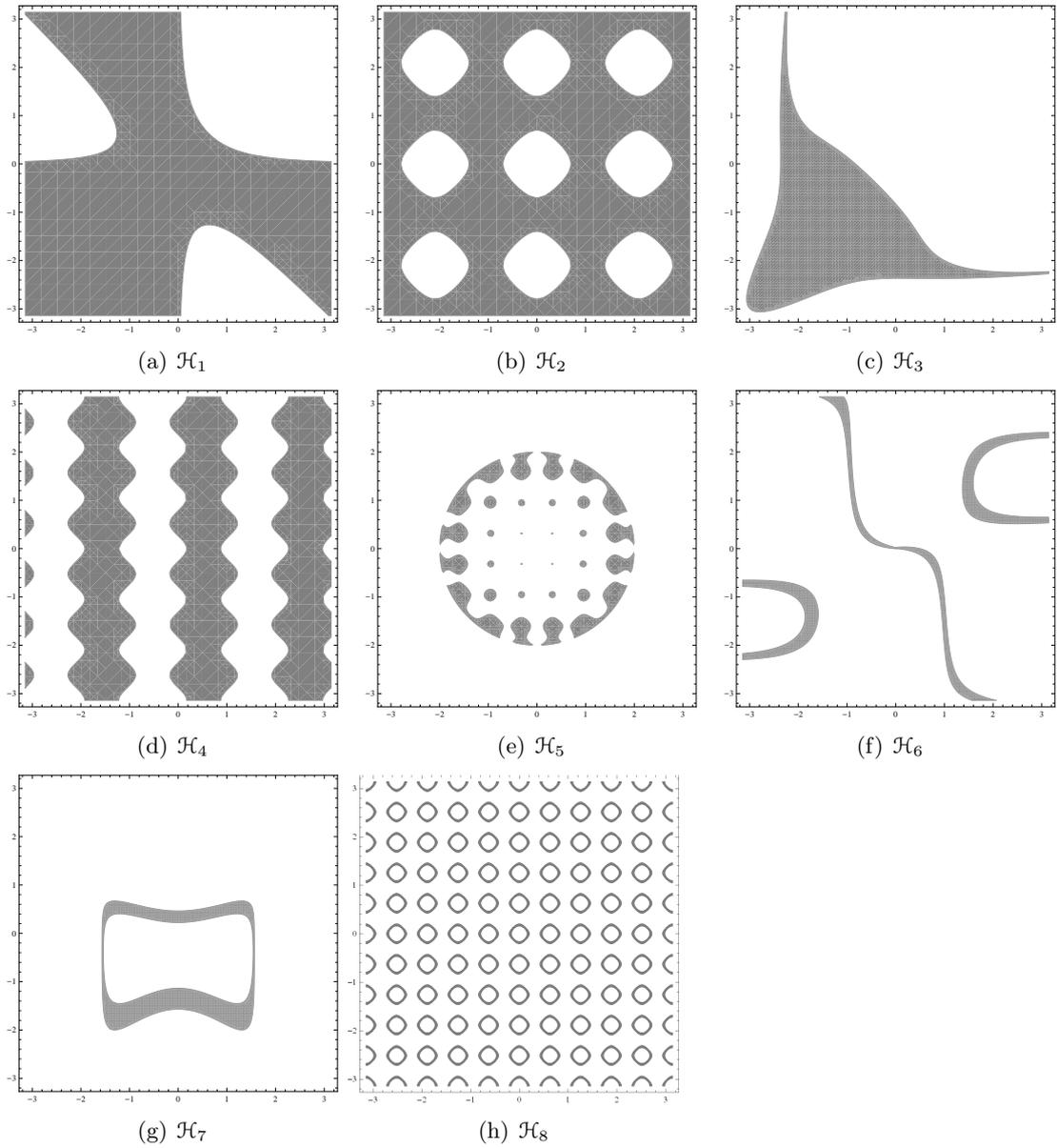
$C_1$	$x^2y + y^2x \leq 0.5$	$C_{\mathcal{H}_1} = \{C_1\}$
$C_2$	$\cos(3x) + \cos(3y) \leq 0.5$	$C_{\mathcal{H}_2} = \{C_2\}$
$C_3, C_4$	$-1.5 \leq e^{x+1.5}(y+1.5)^3 + e^{y+1.5}(x+1.5)^3 \leq 8$	$C_{\mathcal{H}_3} = \{C_3, C_4\}$
$C_5$	$\sin(3x) + \sin(3y)^2 \geq 0.5$	$C_{\mathcal{H}_4} = \{C_5\}$
$C_6, C_7$	$10\cos(5x)^2 + 10\cos(5y)^2 \leq (x^2 + y^2)^2, x^2 + y^2 \leq 4$	$C_{\mathcal{H}_5} = \{C_6, C_7\}$
$C_8, C_9$	$0 \leq 2x + y - 5 \log(x^2)\sin(y) \leq 1$	$C_{\mathcal{H}_6} = \{C_8, C_9\}$
$C_{10}, C_{11}$	$1 \leq \cos(x)(e^{\cos(y)} - \cos(x) - \sin(y))^5 \leq 6$	$C_{\mathcal{H}_7} = \{C_{10}, C_{11}\}$
$C_{12}, C_{13}$	$0.5 \leq \cos(10x) + \cos(10y) \leq 1$	$C_{\mathcal{H}_8} = \{C_{12}, C_{13}\}$

**Table 4.4:** Definition of constraints and events.

### 4.8.2 The PC Events

Consider the CCSP  $CP$  where  $X = \langle x, y \rangle$ ,  $D = [-\pi, \pi] \times [-\pi, \pi]$  and  $C = \{C_1, \dots, C_{13}\}$ , compactly defined in Table 4.4. Consider two PCs,  $\langle CP, f_1 \rangle$  and  $\langle CP, f_2 \rangle$ , where  $f_1$  is the bi-normal distribution (figure 4.8 (a)) defined over  $\Omega_1 = \mathbb{R}^2$  and  $f_2$  is a custom distribution defined over  $\Omega_2 = D$  (figure 4.8 (b)). Consider the events  $\mathcal{H}_i$ , with  $1 \leq i \leq 8$  (shown in figure 4.9), whose associated subset of constraints,  $C_{\mathcal{H}_i}$ , is defined in the last column of Table 4.4.


**Figure 4.8:** Probability density functions used for testing.



**Figure 4.9:** Events used for testing.

### 4.8.3 Probability Enclosure

Here we present the results of applying algorithm 6 to the set of events defined above and compare them with the approximate ones obtained using *Mathematica*. In this context *Mathematica* is not used to solve a Continuous CSP (i.e. find solutions) but rather to compute the probability of events, and so it can be compared with our approach. For that purpose function *NProbability* is used with the default parametrization. This function relies on function *NIntegrate* with its default global adaptive integration strategy and the integration region is defined by boolean functions that check whether the constraints (associated with the event) are satisfied.

The next tables show the results of applying both versions of algorithm 6 and *Mathematica* to previously defined events, with  $f = f_1$  (table 4.5) and  $f = f_2$  (table 4.6). The PCTM version of the algorithm was parameterized with  $\varepsilon = 10^{-15}$ ,  $\delta = 0.001$  and a Taylor model order of 2 and the PCMC version with  $\varepsilon = 10^{-15}$ ,  $\delta = 0.005$  and  $N = 100$  sample points. The first two columns refer to results of the PCTM version of algorithm 6: the enclosure for the probability of the event,  $[P(\mathcal{H})]$ , and the CPU time to compute it. The next three columns refer to results of the PCMC version of algorithm 6: the numerical approximate probability of the event,  $\hat{P}(\mathcal{H})$ , the CPU time to compute it and a lower bound  $\mathcal{E}$  for the relative error of  $\hat{P}(\mathcal{H})$  based on the safe enclosure  $[P(\mathcal{H})]$ , as follows.

$$\mathcal{E} = \begin{cases} \left( \frac{\hat{P}(\mathcal{H}) - [P(\mathcal{H})]}{[P(\mathcal{H})]} \right) / [P(\mathcal{H})] & \text{if } \hat{P}(\mathcal{H}) \geq [P(\mathcal{H})] \\ \left( \frac{[P(\mathcal{H})] - \hat{P}(\mathcal{H})}{[P(\mathcal{H})]} \right) / [P(\mathcal{H})] & \text{if } \hat{P}(\mathcal{H}) \leq [P(\mathcal{H})] \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

The last three columns refer to the approximate *Mathematica* results with the same meaning as the previous three columns.

The results from tables 4.5 and 4.6 show that significant errors may be incurred in the computation of probabilities when events correspond to highly non-linear constraints even with the classical bi-normal distribution. Whereas *Mathematica* provides approximate values with small errors for events with high probability (with an execution time similar to that used by our safe methods), for events with low probability it provides faster results, but the errors incurred are so important as to make the approximate

$\mathcal{H}$	<i>PCTM</i>		<i>PCMC</i>			<i>Mathematica</i>		
	$[P(\mathcal{H})]$	T (s)	$\widehat{P}(\mathcal{H})$	T (s)	$\varepsilon$ (%)	$\widehat{P}(\mathcal{H})$	T (s)	$\varepsilon$ (%)
$\mathcal{H}_1$	[0.8044,0.8054]	5	0.8051	3	0.00	0.8051	1	0.00
$\mathcal{H}_2$	[0.6798,0.6808]	6	0.6807	5	0.00	0.6809	23	0.01
$\mathcal{H}_3$	[0.2542,0.2552]	5	0.2549	1	0.00	0.2525	29	0.66
$\mathcal{H}_4$	[0.4977,0.4987]	7	0.4985	3	0.00	0.4466	24	10.25
$\mathcal{H}_5$	[0.1105,0.1115]	71	0.1114	3	0.00	0.0870	27	21.31
$\mathcal{H}_6$	[0.0566,0.0576]	20	0.0576	1	0.00	0.0349	15	38.26
$\mathcal{H}_7$	[0.1356,0.1366]	65	0.1370	1	0.29	0.0773	26	43.02
$\mathcal{H}_8$	[0.1223,0.1233]	185	0.1235	17	0.16	0.0048	15	96.04

**Table 4.5:** Probability enclosures for events  $\mathcal{H}_i$  with PDF  $f_1$ , obtained with both versions of algorithm 6, their numerical computations with *Mathematica* and respective timings.

$\mathcal{H}$	<i>PCTM</i>		<i>PCMC</i>			<i>Mathematica</i>		
	$[P(\mathcal{H})]$	T (s)	$\widehat{P}(\mathcal{H})$	T (s)	$\varepsilon$ (%)	$\widehat{P}(\mathcal{H})$	T (s)	$\varepsilon$ (%)
$\mathcal{H}_1$	[0.6557,0.6567]	10	0.6564	4	0.00	0.6565	2	0.00
$\mathcal{H}_2$	[0.6910,0.6920]	10	0.6918	12	0.00	0.6920	10	0.01
$\mathcal{H}_3$	[0.2589,0.2599]	23	0.2597	2	0.00	0.2471	8	4.58
$\mathcal{H}_4$	[0.4993,0.5003]	12	0.5001	8	0.00	0.3305	9	33.81
$\mathcal{H}_5$	[0.0563,0.0573]	31	0.0573	1	0.00	0.0401	10	28.87
$\mathcal{H}_6$	[0.0588,0.0598]	25	0.0583	1	0.85	0.0173	10	70.48
$\mathcal{H}_7$	[0.0864,0.0874]	36	0.0876	1	0.23	0.0239	10	72.36
$\mathcal{H}_8$	[0.1228,0.1238]	279	0.1238	25	0.00	0.0085	9	93.08

**Table 4.6:** Probability enclosures for events  $\mathcal{H}_i$  with PDF  $f_2$ , obtained with both versions of algorithm 6, their numerical computations with *Mathematica* and respective timings.

probabilities largely useless (e.g. for cost-benefit analysis where events associated to the malfunctioning of a system are expected to be rare).

Several combinations of  $\delta$  and  $\varepsilon$  can be made. Here, we chose a very small  $\varepsilon$  so the algorithm stops when the required accuracy ( $\delta$ ) for the probability is achieved (a larger  $\varepsilon$  could cause the algorithm to stop without reaching that accuracy). Choosing a smaller  $\delta$  (maintaining  $\varepsilon$ ) would increase the computation time and, evidently, the accuracy of the probability enclosure. In the events presented here, higher order degree approximations take longer to compute, but the increase in accuracy is not significant.

#### 4.8.4 Conditional Probability Enclosure

The results of applying version PCTM of algorithm 7 to the set of events in section 4.8.2 with the bi-normal PDF  $f_1$  and its comparison with those obtained with separate computations of the joint box covers are now presented. Furthermore these results are compared with the approximate ones obtained with *Mathematica*.

Table 4.7 shows, for each event  $\mathcal{H}_i$ , the midpoint of the enclosure for the conditional probability of every other event given  $\mathcal{H}_i$ ,  $[P(\mathcal{H}|\mathcal{H}_i)]$ , and the CPU time, in seconds, to compute such enclosure using PCTM version of algorithm 7, parameterized with  $\varepsilon = 10^{-15}$ ,  $\delta = 0.001$  and a Taylor model order of 2. Table 4.8 refers to the results  $\hat{P}(\mathcal{H}|\mathcal{H}_i)$  obtained with *Mathematica*, showing a lower bound for the relative error of  $\hat{P}(\mathcal{H}|\mathcal{H}_i)$  based on the safe enclosure  $[P(\mathcal{H}|\mathcal{H}_i)]$  computed as in formula (4.9).

$\mathcal{H}$	$[P(\mathcal{H} \mathcal{H}_1)]$		$[P(\mathcal{H} \mathcal{H}_2)]$		$[P(\mathcal{H} \mathcal{H}_3)]$		$[P(\mathcal{H} \mathcal{H}_4)]$		$[P(\mathcal{H} \mathcal{H}_5)]$		$[P(\mathcal{H} \mathcal{H}_6)]$		$[P(\mathcal{H} \mathcal{H}_7)]$	
	<i>mid</i>	T	<i>mid</i>	T	<i>mid</i>	T								
$\mathcal{H}_1$			0.7790	32	0.7722	46	0.6605	46	0.6983	785	0.6827	452	0.7052	547
$\mathcal{H}_2$	0.6584	27			0.8926	32	0.7453	49	0.6066	727	0.7878	430	0.7633	569
$\mathcal{H}_3$	0.2825	10	0.2892	19			0.2213	16	0.3570	381	0.9890	77	0.3089	255
$\mathcal{H}_4$	0.4613	28	0.4837	31	0.4327	34			0.5135	710	0.4508	269	0.5860	431
$\mathcal{H}_5$	0.8360	61	0.1140	67	0.1556	67	0.1144	55			0.1531	129	0.1526	239
$\mathcal{H}_6$	0.5590	19	0.5730	25	0.2200	10	0.5170	14	0.7890	147			0.3580	52
$\mathcal{H}_7$	0.1039	71	0.1412	71	0.1651	58	0.1602	63	0.1872	404	0.8540	58		

**Table 4.7:** Midpoints of the conditional probability enclosures for  $\mathcal{H}$  given  $\mathcal{H}_i$  obtained by the PCTM version of algorithm 7 and respective timings (in seconds).

$\mathcal{H}$	$\mathcal{E}(\mathcal{H} \mathcal{H}_1)$	$\mathcal{E}(\mathcal{H} \mathcal{H}_2)$	$\mathcal{E}(\mathcal{H} \mathcal{H}_3)$	$\mathcal{E}(\mathcal{H} \mathcal{H}_4)$	$\mathcal{E}(\mathcal{H} \mathcal{H}_5)$	$\mathcal{E}(\mathcal{H} \mathcal{H}_6)$	$\mathcal{E}(\mathcal{H} \mathcal{H}_7)$
$\mathcal{H}_1$		0.75 %	0.12 %	1.17 %	5.33 %	42.88 %	21.31 %
$\mathcal{H}_2$	0.68 %		0.59 %	2.80 %	6.81 %	24.95 %	41.01 %
$\mathcal{H}_3$	0.10 %	0.98 %		57.07 %	29.86 %	100.00 %	100.00 %
$\mathcal{H}_4$	12.84 %	11.44 %	61.21 %		91.60 %	92.34 %	4.09 %
$\mathcal{H}_5$	26.59 %	25.64 %	44.49 %	92.64 %		65.02 %	100.00 %
$\mathcal{H}_6$	22.82 %	64.81 %	100.00 %	94.73 %	72.64 %		100.00 %
$\mathcal{H}_7$	58.28 %	55.26 %	100.00 %	33.87 %	100.00 %	100.00 %	

**Table 4.8:** Relative error percentages of the conditional probabilities of  $\mathcal{H}$  given  $\mathcal{H}_i$  obtained with *Mathematica* when compare with those computed by algorithm 7.

The results from Tables 4.7 and 4.8 regarding the computation of conditional probabilities confirm the findings for the unconditional case. As before, the errors incurred by *Mathematica* with the computed approximate solutions are more significant when the events have low probability. But now the situation is more complex since a conditional probability can be high but resulting from the division of two small probabilities. Hence, significant errors are now observed in relatively large conditional probabilities (e.g. in the case of  $P(\mathcal{H}_6|\mathcal{H}_2) \approx 0.5730$ , *Mathematica* computes a probability with an error of 64.81%). Moreover, in some cases the error is 100%, in which case *Mathematica* reports that (possible) events may never occur, which is an extreme case of their useless value regarding cost-benefit analysis.

Table 4.9 refers to the gains ( $g$ ) obtained with algorithm 7 when compared with the alternative method (with algorithm 6) that uses two completely independent joint box covers, one for  $\mathcal{H} \cap \mathcal{H}_i$  and other for  $\mathcal{H}_i$ . The comparison was based on the total number of boxes of the computed covers which is proportional to the CPU time.

$\mathcal{H}$	$g(\mathcal{H} \mathcal{H}_1)$	$g(\mathcal{H} \mathcal{H}_2)$	$g(\mathcal{H} \mathcal{H}_3)$	$g(\mathcal{H} \mathcal{H}_4)$	$g(\mathcal{H} \mathcal{H}_5)$	$g(\mathcal{H} \mathcal{H}_6)$	$g(\mathcal{H} \mathcal{H}_7)$
$\mathcal{H}_1$		33.99 %	28.46 %	32.20 %	40.78 %	42.08 %	40.45 %
$\mathcal{H}_2$	25.28 %		35.13 %	35.40 %	40.60 %	42.53 %	40.85 %
$\mathcal{H}_3$	10.45 %	20.77 %		25.02 %	32.07 %	24.77 %	30.37 %
$\mathcal{H}_4$	24.93 %	24.75 %	19.62 %		35.38 %	31.69 %	36.22 %
$\mathcal{H}_5$	6.76 %	7.05 %	0.60 %	6.54 %		21.10 %	26.00 %
$\mathcal{H}_6$	2.80 %	2.79 %	2.04 %	6.04 %	9.08 %		6.08 %
$\mathcal{H}_7$	9.05 %	8.02 %	7.40 %	6.68 %	13.26 %	19.25 %	

**Table 4.9:** Speedup of algorithm 7 when compared with the alternative method (with algorithm 6) that uses two completely independent joint box covers.

While the computation time of safe enclosures for the conditional probability of events typically doubles that of unconditional probabilities (as it requires the division of two such values) algorithm 7 minimizes such effect, by speeding up the computations by around 25%, as shown in Table 4.9.

## 4.9 Summary

In this chapter the definition and the semantics of probabilistic continuous constraint space, the core of the proposed probabilistic continuous constraint framework, was pre-

sented and the kind of problems that can be formulated within this probabilistic space were identified. The concept of probabilistic constraint event was introduced and both safe and approximate methods to obtain enclosures for their unconditional and conditional probabilities were described. Experimental results illustrated the capabilities of the proposed algorithms.

The next chapter presents methods to compute safe and approximate enclosures for conditional or unconditional probabilistic features of random vectors within a probabilistic continuous constraint space, and to provide their probability distributions.

## CHAPTER 5

# Random Vectors

Once addressed the probability of PC events in the previous chapter, we now discuss how to compute enclosures for probabilistic features of random vectors (RV) in unconditional (section 5.1) and conditional probability spaces (section 5.2), in the scope of the Probabilistic Constraint framework. Section 5.3 presents algorithms to compute the enclosures discussed in the two previous sections. Then, Section 5.4 describes how to compute non parametric probability distributions of random vectors. All definitions and properties presented in this chapter assume safe enclosures for multidimensional integrals, except in section 5.5 that discusses the impact of adopting an approximate method to compute such enclosures. Finally, section 5.6 shows and discusses the results of applying the proposed algorithms to a set of benchmarks.

The definitions and properties that follow consider a PC  $\langle\langle X, D, C \rangle, f\rangle$ , where  $D \in \mathbb{IR}^n$ . For the sake of generality, in the following we consider  $\mathbf{X}$  to be the identity random vector associated with the PC probability space and  $\mathbf{Y} = \langle Y_1, \dots, Y_m \rangle$  to be a generic random vector where each  $Y_i$  is as a function of  $\mathbf{X}$ <sup>1</sup>, i.e.  $Y_i = g_i(\mathbf{X})$ , defined on the same probability space.

## 5.1 Probabilistic Enclosures

We start with enclosures for the probability of random vectors' values as well as expected values and the covariance matrices of random vectors.

---

<sup>1</sup>See property 3.10

Notice that we are only able to compute enclosures for expected values of random vectors (and consequently for covariance matrices) when  $\Omega$  corresponds exactly to  $D$ , i.e., when  $\Omega$  is bounded. This is due to the limitations of the proposed integration methods to compute improper integrals. When  $\Omega$  is unbounded, expected values for random vectors are computed, necessarily, conditioned by  $D$ .

Since restricting the range of values of a random vector is an event, an enclosure for the probability that a random vector takes certain values can be obtained directly from property 4.5, as follows.

**Property 5.1 (Enclosure for the Probability of a RV)** *Given any bounded box  $B_{\mathbf{Y}} = I_1 \times \cdots \times I_m \in \mathbb{IR}^m$ , the probability that random vector  $\mathbf{Y}$  takes values in  $B_{\mathbf{Y}}$ ,  $P(\mathbf{Y} \in B_{\mathbf{Y}}) = P(Y_1 \in I_1, \dots, Y_m \in I_m)$ , is  $P(\mathcal{H})$  where  $\mathcal{H}$  is the PC event:*

$$\mathcal{H} = \mathcal{F}(\langle X, D, \{\underline{I}_1 \leq g_1(\mathbf{X}) \leq \overline{I}_1, \dots, \underline{I}_m \leq g_m(\mathbf{X}) \leq \overline{I}_m\} \rangle)^1$$

*An enclosure for the probability  $P(\mathcal{H})$  can be computed as in property 4.5.*

**Proof.** It follows from definition 3.20, that addresses events induced via random vectors. ■

This property can be illustrated as follows.

**Example 5.1.**

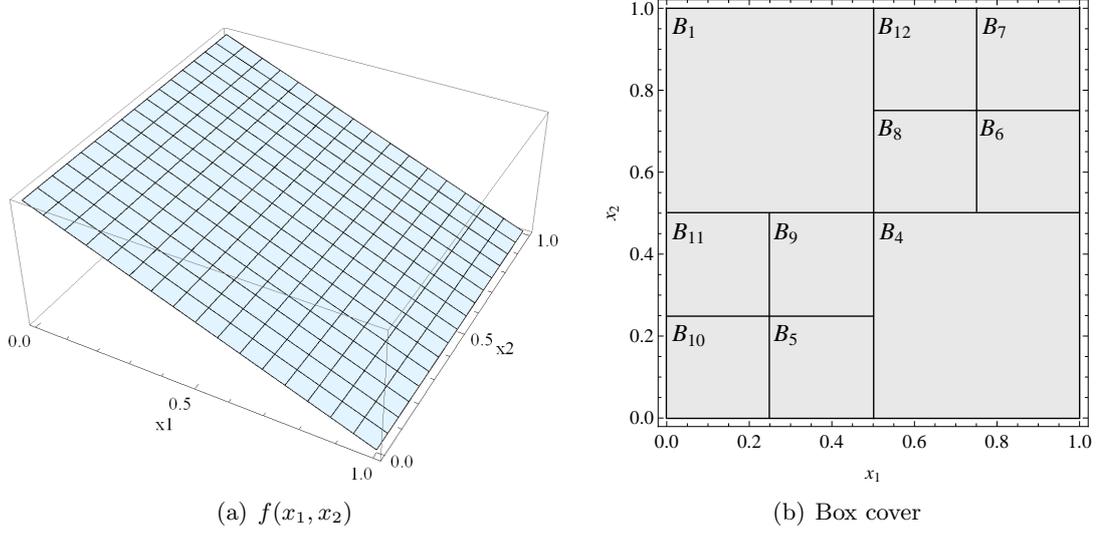
Consider PC  $\langle \langle X = \langle x_1, x_2 \rangle, D = [0, 1] \times [0, 1], C \rangle, f \rangle$ , where  $f(x_1, x_2) = 2 - 2x_1$  (shown in figure 5.1 (a)),  $\Omega = D$  and the associated random vector  $\mathbf{X} = \langle X_1, X_2 \rangle$  and a box cover<sup>2</sup>  $\mathcal{D}_{\square} = \{B_1\} \cup \{B_i : 4 \leq i \leq 12\}$  of  $D$  (shown in figure 5.1 (b)).

Table 5.1 shows enclosures for the probabilities of the boxes in  $\mathcal{D}_{\square}$ , computed as in property 4.4. While the first column considers the boxes to belong to an inner joint box cover, the second column considers them to be boundary boxes<sup>3</sup>.

<sup>1</sup>Notice that  $\underline{I}_i \leq g_i(\mathbf{X}) \leq \overline{I}_i$  is an abbreviated way to represent the two constraints whose relations are defined by:  $\rho_{i_1} = \{d \in D : I_i - g_i(d) \leq 0\}$  and  $\rho_{i_2} = \{d \in D : g_i(d) - \overline{I}_i \leq 0\}$ .

<sup>2</sup>The indices were maintained from the examples of chapter 4.

<sup>3</sup>We chose to simplify the very sharp enclosures obtained for the rational numbers (with 16 digits precision) with a high order Taylor model, by showing the corresponding fraction. Likewise, in the cruder computations, the obtained rational numbers are shown as fractions.



**Figure 5.1:** PDF and box cover used in examples.

Figure 5.2 shows the events  $\mathcal{H}_6$ ,  $\mathcal{H}_7$ ,  $\mathcal{H}_8$  and  $\mathcal{H}_9$  resulting from restricting the values of random vectors to boxes. In the first three figures the random vector is  $\mathbf{Y} = \langle X_1, X_2 \rangle$ , while in the last it is  $\mathbf{Y} = \langle X_1 - X_2 \rangle$ .

Given the joint box covers:

$$\begin{aligned} \mathcal{H}_{\boxplus_6} &= \langle \{B_4, B_6, B_7\}, \{B_6, B_7\} \rangle \\ \mathcal{H}_{\boxplus_7} &= \langle \{B_1, B_6, B_7, B_8, B_{12}\}, \{B_1, B_6, B_7, B_8, B_{12}\} \rangle \\ \mathcal{H}_{\boxplus_8} &= \langle \{B_4, B_5, B_9\}, \{B_5, B_9\} \rangle \\ \mathcal{H}_{\boxplus_9} &= \langle \{B_i : i \in \{4, \dots, 10\}\}, \{B_5, B_6\} \rangle \end{aligned}$$

for events, respectively,  $\mathcal{H}_6$ ,  $\mathcal{H}_7$ ,  $\mathcal{H}_8$  and  $\mathcal{H}_9$ , table 5.2 shows probability enclosures of those events over the boxes  $B_i$  (using directly the values of table 5.1).

With  $\mathbf{Y} = \langle X_1, X_2 \rangle$  (i.e.  $g_i(\mathbf{X}) = X_i$ ) and using the values of table 5.2 and property 4.5, we obtain enclosures for the probability of events  $\mathcal{H}_6$ ,  $\mathcal{H}_7$  and  $\mathcal{H}_8$  using, respectively, the joint box covers  $\mathcal{H}_{\boxplus_6}$ ,  $\mathcal{H}_{\boxplus_7}$  and  $\mathcal{H}_{\boxplus_8}$  are:

$$\begin{aligned} P(X_1 \geq 0.75) &= P(\mathcal{H}_6) = \frac{2}{32} \in [P](\mathcal{H}_{\boxplus_6}, f) = \left[ \frac{1}{32}, \frac{9}{32} \right] \\ P(X_2 \geq 0.5) &= P(\mathcal{H}_7) = \frac{1}{2} \in [P](\mathcal{H}_{\boxplus_7}, f) = \left[ \frac{1}{2} \right] \end{aligned}$$

$B$	$[I](f, B)$	$[0, \sup([f](B)) * \text{vol}(B)]$
$B_1$	$[3/8]$	$[0, 1/2]$
$B_4$	$[1/8]$	$[0, 1/4]$
$B_5$	$[5/64]$	$[0, 3/32]$
$B_6$	$[1/64]$	$[0, 1/32]$
$B_7$	$[1/64]$	$[0, 1/32]$
$B_8$	$[3/64]$	$[0, 1/16]$
$B_9$	$[5/64]$	$[0, 3/32]$
$B_{10}$	$[7/64]$	$[0, 1/8]$
$B_{11}$	$[7/64]$	$[0, 1/8]$
$B_{12}$	$[3/64]$	$[0, 1/16]$

**Table 5.1:** Probability enclosures of  $B_i$  using property 4.4.

$B$	$[P_{\mathcal{H}_6}](f, B)$	$[P_{\mathcal{H}_7}](f, B)$	$[P_{\mathcal{H}_8}](f, B)$	$[P_{\mathcal{H}_9}](f, B)$
$B_1$		$[3/8]$		
$B_4$	$[0, 1/4]$		$[0, 1/4]$	$[0, 1/4]$
$B_5$			$[5/64]$	$[5/64]$
$B_6$	$[1/64]$	$[1/64]$		$[1/64]$
$B_7$	$[1/64]$	$[1/64]$		$[0, 1/32]$
$B_8$		$[3/64]$		$[0, 1/16]$
$B_9$			$[5/64]$	$[0, 3/32]$
$B_{10}$				$[0, 1/8]$
$B_{12}$		$[3/64]$		

**Table 5.2:** Enclosures for the probability of  $B_i \cap \mathcal{H}_j$ .

$$P(0.25 \leq X_1 \leq 0.75 \wedge X_2 \leq 0.5) = P(\mathcal{H}_8) = \frac{8}{32} \in [P](\mathcal{H}_{\boxplus 8}, f) = \left[ \frac{5}{32}, \frac{13}{32} \right]$$

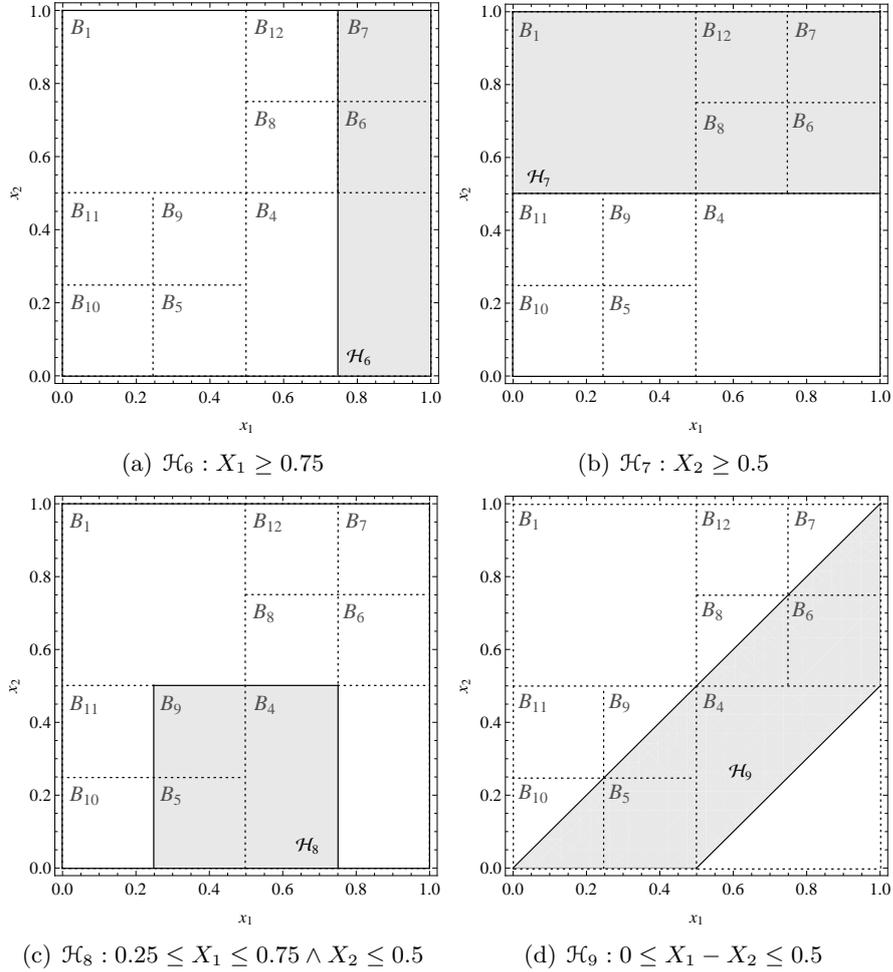
Now with  $\mathbf{Y} = \langle X_1 - X_2 \rangle$  (i.e.,  $g_1(\mathbf{X}) = X_1 - X_2$ ) and using the values of table 5.2 and property 4.5, an enclosure for the probability of  $\mathcal{H}_9$  using the joint box cover  $\mathcal{H}_{\boxplus 9}$  is:

$$P(0 \leq Y_1 \leq 0.5) = P(\mathcal{H}_9) = \frac{7}{24} \in [P](\mathcal{H}_{\boxplus 9}, f) = \left[ \frac{3}{32}, \frac{21}{32} \right]$$

The exact probability values were obtained analytically.

---

Obtaining the expected value of a random variable  $Y_i = g_i(\mathbf{X})$  with respect to the joint PDF  $f(\mathbf{x})$  of  $\mathbf{X}$  implies the computation of a multidimensional integral of  $g_i(\mathbf{x})f(\mathbf{x})$



**Figure 5.2:** Restricting the values of random variables.

over a box (i.e., the Cartesian product of the ranges of random variables  $X_i$ ). A safe enclosure for such value can be computed as in definition 3.36. The integration is not made directly over  $D$ , but rather on a partition  $\mathcal{H}_\square$  of  $D$ , since this allows tighter enclosures to be obtained (see property 3.17).

**Property 5.2 (Enclosure for the Expected Value of a RV)** *Given a joint box cover  $\mathcal{D}_\boxplus = \langle \mathcal{D}_\square, \mathcal{D}_\blacksquare \rangle$  of  $\mathcal{D} = \mathcal{F}(\langle X, D, \{\} \rangle)$ , where  $\mathcal{D}_\square = \mathcal{D}_\blacksquare$  and  $D = \Omega$ , an enclosure for the expected value of  $\mathbf{Y}$  is given by:*

$$[E](\mathbf{Y}, \mathcal{D}_\boxplus, f) = \langle [E](Y_1, \mathcal{D}_\boxplus, f), \dots, [E](Y_m, \mathcal{D}_\boxplus, f) \rangle$$

where,  $\forall 1 \leq i \leq m$

$$E[Y_i] \in [E](Y_i, \mathcal{D}_{\boxplus}, f) = \sum_{B \in \mathcal{D}_{\square}} [I](g_i f, B)$$

**Proof.** Definition 3.29 states that  $E[g_i(\mathbf{X})] = \int_{\mathbb{R}^n} g_i(\mathbf{x})f(\mathbf{x})d\mathbf{x}$  and the computation of the expected value implies the computation of an integral over an unbounded box  $\mathbb{R}^n$ . Nevertheless, since we assume  $\Omega = D$ , then  $\Omega$  is a bounded box and the joint PDF  $f$  is defined over  $D$  and 0 elsewhere. Then

$$\begin{aligned} \int_{\mathbb{R}^n} g_i(\mathbf{x})f(\mathbf{x})d\mathbf{x} &= \int_{\Omega} g_i(\mathbf{x})f(\mathbf{x})d\mathbf{x} + \int_{\mathbb{R}^n \setminus \Omega} g_i(\mathbf{x})f(\mathbf{x})d\mathbf{x} \\ &= \int_{\Omega} g_i(\mathbf{X})f(\mathbf{x})d\mathbf{x} \end{aligned}$$

since  $\int_{\mathbb{R}^n \setminus \Omega} g_i(\mathbf{x})f(\mathbf{x})d\mathbf{x} = 0$ .

Finally, by property 3.17,  $\int_{\mathbb{R}^n} g_i(\mathbf{x})f(\mathbf{x})d\mathbf{x} \in \sum_{B \in \mathcal{D}_{\square}} [I](g_i f, B)$ . ■

Adopting the expression for variance in property 3.11, an enclosure for the covariance matrix of a random vector can also be defined.

**Property 5.3 (Enclosure for the Covariance Matrix of a RV)** Given a joint box cover  $\mathcal{D}_{\boxplus}$  of  $\mathcal{D} = \mathcal{F}(\langle X, D, \{\} \rangle)$ , where  $\mathcal{D}_{\square} = \mathcal{D}_{\blacksquare}$  and  $D = \Omega$ , an enclosure for the covariance matrix  $\Sigma$  of  $\mathbf{Y}$  is given by:

$$\begin{aligned} \Sigma_{ij}[\mathbf{Y}] = Cov(Y_i, Y_j) &\in [\Sigma_{ij}](\mathbf{Y}, \mathcal{D}_{\boxplus}, f) = [Cov](Y_i, Y_j, \mathcal{D}_{\boxplus}, f) \\ &= [E](Y_i Y_j, \mathcal{D}_{\boxplus}, f) - [E](Y_i, \mathcal{D}_{\boxplus}, f)[E](Y_j, \mathcal{D}_{\boxplus}, f) \end{aligned}$$

In particular,  $\Sigma_{ii}[\mathbf{Y}] = Var[Y_i] \in [Var](Y_i, \mathcal{H}_{\boxplus}, f) = [E](Y_i^2, \mathcal{D}_{\boxplus}, f) - [E]^2(Y_i, \mathcal{D}_{\boxplus}, f)$ <sup>1</sup>.

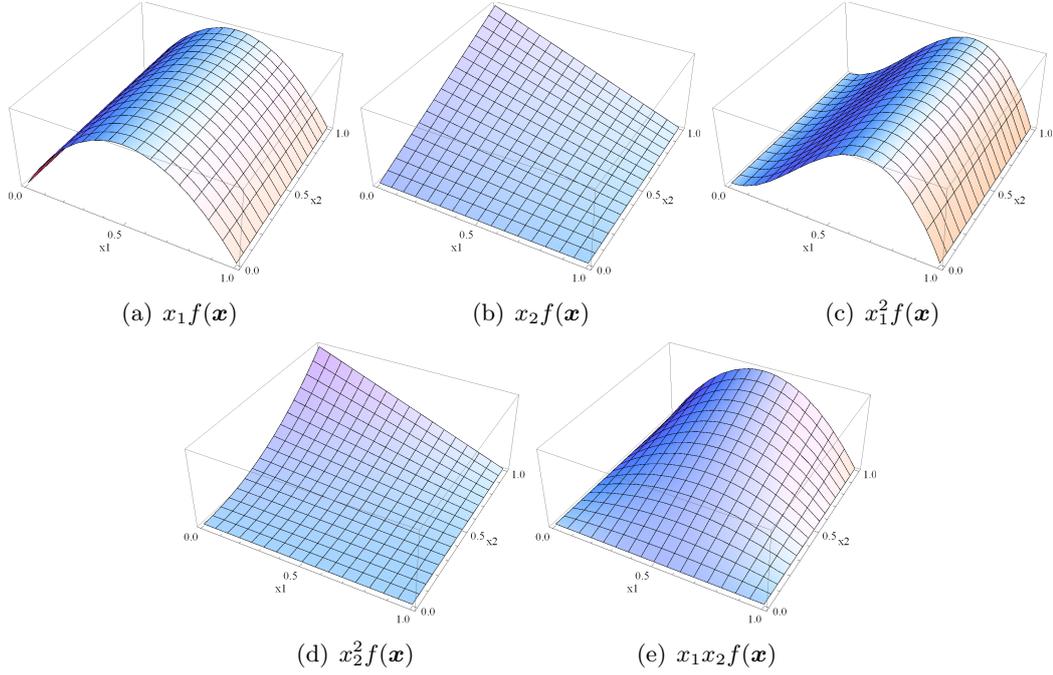
---

<sup>1</sup>Since the variance of a random variable is always  $\geq 0$  the resulting interval is intersected with  $[0, +\infty[$ .

**Proof.** From property 5.2,  $E[Y_i Y_j] \in [E](Y_i Y_j, \mathcal{D}_{\boxplus}, f)$ ,  $E[Y_i] \in [E](Y_i, \mathcal{D}_{\boxplus}, f)$  and  $E[Y_j] \in [E](Y_j, \mathcal{D}_{\boxplus}, f)$ . By the properties of interval arithmetic

$$\text{Cov}(Y_i, Y_j) = E[Y_i Y_j] - E[Y_i]E[Y_j] \in [E](Y_i Y_j, \mathcal{D}_{\boxplus}, f) - [E](Y_i, \mathcal{D}_{\boxplus}, f)[E](Y_j, \mathcal{D}_{\boxplus}, f)$$

■



**Figure 5.3:** Integrand functions.

**Example 5.2.** Consider the PC of example 5.1, the random vector  $\mathbf{Y} = \langle X_1, X_2 \rangle$  (i.e.  $g_i(\mathbf{X}) = X_i$ ) and the box covers  $\mathcal{D}_{\square} = \mathcal{D}_{\blacksquare} = \{B_i : i \in \{4, \dots, 12\}\} \cup \{B_1\}$  ( $B_i$  are the boxes in figure 5.1 (b)). Let  $\mathcal{D}_{\boxplus} = \langle \mathcal{D}_{\square}, \mathcal{D}_{\blacksquare} \rangle$  be a joint box cover of  $\mathcal{D} = \mathcal{F}(\langle X, D, \{\} \rangle)$ .

Table 5.3 shows very crude enclosures for integrals of several integrand functions (shown in figure 5.3) over boxes, computed with Taylor models of order 0. Using such values, an enclosure for the expected value of  $\mathbf{Y}$  is:

$$\begin{aligned} [E](\mathbf{Y}, \mathcal{D}_{\boxplus}, f) &= \left\langle \sum_{B \in \mathcal{D}_{\square}} [I](x_1 f, B), \sum_{B \in \mathcal{D}_{\square}} [I](x_2 f, B) \right\rangle \\ &= \left\langle \left[ \frac{1}{16}, \frac{13}{16} \right], \left[ \frac{13}{64}, \frac{61}{64} \right] \right\rangle, \end{aligned}$$

$B$	$[I](x_1 f, B)$	$[I](x_1^2 f, B)$	$[I](x_2 f, B)$	$[I](x_2^2 f, B)$	$[I](x_1 x_2 f, B)$
$B_1$	$[0, \frac{1}{4}]$	$[0, \frac{1}{8}]$	$[\frac{1}{8}, \frac{1}{2}]$	$[\frac{1}{16}, \frac{1}{2}]$	$[0, \frac{1}{4}]$
$B_4$	$[0, \frac{1}{4}]$	$[0, \frac{1}{4}]$	$[0, \frac{1}{8}]$	$[0, \frac{1}{16}]$	$[0, \frac{1}{8}]$
$B_5$	$[\frac{1}{64}, \frac{3}{64}]$	$[\frac{1}{256}, \frac{3}{128}]$	$[0, \frac{3}{128}]$	$[0, \frac{3}{512}]$	$[0, \frac{3}{256}]$
$B_6$	$[0, \frac{1}{32}]$	$[0, \frac{1}{32}]$	$[0, \frac{3}{128}]$	$[0, \frac{9}{512}]$	$[0, \frac{3}{128}]$
$B_7$	$[0, \frac{1}{32}]$	$[0, \frac{1}{32}]$	$[0, \frac{1}{32}]$	$[0, \frac{1}{32}]$	$[0, \frac{1}{32}]$
$B_8$	$[\frac{1}{64}, \frac{3}{64}]$	$[\frac{1}{128}, \frac{9}{256}]$	$[\frac{1}{64}, \frac{3}{64}]$	$[\frac{1}{128}, \frac{9}{256}]$	$[\frac{1}{128}, \frac{9}{256}]$
$B_9$	$[\frac{1}{64}, \frac{3}{64}]$	$[\frac{1}{256}, \frac{3}{128}]$	$[\frac{1}{64}, \frac{3}{64}]$	$[\frac{1}{256}, \frac{3}{128}]$	$[\frac{1}{256}, \frac{3}{128}]$
$B_{10}$	$[0, \frac{1}{32}]$	$[0, \frac{1}{128}]$	$[0, \frac{1}{32}]$	$[0, \frac{1}{128}]$	$[0, \frac{1}{128}]$
$B_{11}$	$[0, \frac{1}{32}]$	$[0, \frac{1}{128}]$	$[\frac{3}{128}, \frac{1}{16}]$	$[\frac{3}{512}, \frac{1}{32}]$	$[0, \frac{1}{64}]$
$B_{12}$	$[\frac{1}{64}, \frac{3}{64}]$	$[\frac{1}{128}, \frac{9}{256}]$	$[\frac{3}{128}, \frac{1}{16}]$	$[\frac{9}{512}, \frac{1}{16}]$	$[\frac{3}{256}, \frac{3}{64}]$

**Table 5.3:** Enclosures for the integrals  $\int_B x_j f(\mathbf{x}) d\mathbf{x}$ ,  $\int_B x_j^2 f(\mathbf{x}) d\mathbf{x}$  and  $\int_B x_1 x_2 f(\mathbf{x}) d\mathbf{x}$ .

an enclosure for the expected values of  $\langle X_1^2, X_2^2 \rangle$  is:

$$\begin{aligned}
 [E](\langle X_1^2, X_2^2 \rangle, \mathcal{D}_{\boxplus}, f) &= \left\langle \sum_{B \in \mathcal{D}_{\square}} [I](x_1^2 f, B), \sum_{B \in \mathcal{D}_{\square}} [I](x_2^2 f, B) \right\rangle \\
 &= \left\langle \left[ \frac{3}{128}, \frac{73}{128} \right], \left[ \frac{25}{256}, \frac{199}{256} \right] \right\rangle,
 \end{aligned}$$

an enclosure for the expected value of  $X_1 X_2$  is:

$$[E](X_1 X_2, \mathcal{D}_{\boxplus}, f) = \sum_{B \in \mathcal{D}_{\square}} [I](x_1 x_2 f, B) = \left[ \frac{3}{128}, \frac{73}{128} \right]$$

and an enclosure for the covariance matrix of  $\mathbf{Y}$  is:

$$\begin{aligned}
 [\Sigma](\mathbf{Y}, \mathcal{D}_{\boxplus}, f) &= \begin{bmatrix} [Var](X_1, \mathcal{D}_{\boxplus}, f) & [Cov](X_1, X_2, \mathcal{D}_{\boxplus}, f) \\ [Cov](X_2, X_1, \mathcal{D}_{\boxplus}, f) & [Var](X_2, \mathcal{D}_{\boxplus}, f) \end{bmatrix} \\
 &= \begin{bmatrix} [0, \frac{145}{256}] & [-\frac{145}{1024}, \frac{577}{1024}] \\ [-\frac{145}{1024}, \frac{577}{1024}] & [\frac{231}{4096}, \frac{351}{4096}] \end{bmatrix}
 \end{aligned}$$

We can observe that, even without uncertainty in the integration region (which is

exactly covered by  $\mathcal{D}_\square$ ), there is uncertainty in the computed quantities due to the source of uncertainty in the computation of the integrals (intentionally high in this example).

Table 5.4 shows the results of computing analytically the integrals involved in the above computations (which are included in the corresponding intervals of table 5.3). The correct values for the expected values and covariance matrix are then computed from those values, and shown to be within the safe bounds computed previously.

$B$	$\int_B x_1 f(\mathbf{x}) d\mathbf{x}$	$\int_B x_1^2 f(\mathbf{x}) d\mathbf{x}$	$\int_B x_2 f(\mathbf{x}) d\mathbf{x}$	$\int_B x_2^2 f(\mathbf{x}) d\mathbf{x}$	$\int_B x_1 x_2 f(\mathbf{x}) d\mathbf{x}$
$B_1$	$\frac{1}{12}$	$\frac{5}{192}$	$\frac{9}{32}$	$\frac{7}{32}$	$\frac{1}{16}$
$B_4$	$\frac{1}{12}$	$\frac{11}{192}$	$\frac{1}{32}$	$\frac{1}{96}$	$\frac{1}{48}$
$B_5$	$\frac{11}{384}$	$\frac{67}{6144}$	$\frac{5}{512}$	$\frac{5}{3072}$	$\frac{11}{3072}$
$B_6$	$\frac{5}{384}$	$\frac{67}{6144}$	$\frac{5}{512}$	$\frac{19}{3072}$	$\frac{25}{3072}$
$B_7$	$\frac{5}{384}$	$\frac{67}{6144}$	$\frac{7}{512}$	$\frac{37}{3072}$	$\frac{35}{3072}$
$B_8$	$\frac{11}{384}$	$\frac{109}{6144}$	$\frac{15}{512}$	$\frac{19}{1024}$	$\frac{55}{3072}$
$B_9$	$\frac{11}{384}$	$\frac{67}{6144}$	$\frac{15}{512}$	$\frac{35}{3072}$	$\frac{11}{1024}$
$B_{10}$	$\frac{5}{384}$	$\frac{13}{6144}$	$\frac{7}{512}$	$\frac{7}{3072}$	$\frac{5}{3072}$
$B_{11}$	$\frac{5}{384}$	$\frac{13}{6144}$	$\frac{21}{512}$	$\frac{49}{3072}$	$\frac{5}{1024}$
$B_{12}$	$\frac{11}{384}$	$\frac{109}{6144}$	$\frac{21}{512}$	$\frac{37}{1024}$	$\frac{77}{3072}$
$D$	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{6}$

**Table 5.4:** Exact values of  $\int_B x_j f(\mathbf{x}) d\mathbf{x}$ ,  $\int_B x_j^2 f(\mathbf{x}) d\mathbf{x}$  and  $\int_B x_1 x_2 f(\mathbf{x}) d\mathbf{x}$ .

The exact value for the expected value of  $\mathbf{Y}$  is:

$$\begin{aligned}
 E[\mathbf{Y}] &= \left\langle \sum_{B \in \mathcal{D}_\square} \int_B x_1 f(\mathbf{x}) d\mathbf{x}, \sum_{B \in \mathcal{D}_\square} \int_B x_2 f(\mathbf{x}) d\mathbf{x} \right\rangle \\
 &= \left\langle \int_D x_1 f(\mathbf{x}) d\mathbf{x}, \int_D x_2 f(\mathbf{x}) d\mathbf{x} \right\rangle \\
 &= \left\langle \frac{128}{384}, \frac{256}{512} \right\rangle = \left\langle \frac{1}{3}, \frac{1}{2} \right\rangle \in \left\langle \left[ \frac{1}{16}, \frac{13}{16} \right], \left[ \frac{13}{64}, \frac{61}{64} \right] \right\rangle,
 \end{aligned}$$

for the expected values of  $\langle X_1^2, X_2^2 \rangle$  is:

$$E[\langle X_1^2, X_2^2 \rangle] = \left\langle \sum_{B \in \mathcal{D}_\square} \int_B x_1^2 f(\mathbf{x}) d\mathbf{x}, \sum_{B \in \mathcal{D}_\square} \int_B x_2^2 f(\mathbf{x}) d\mathbf{x} \right\rangle$$

$$\begin{aligned}
 &= \left\langle \int_D x_1^2 f(\mathbf{x}) d\mathbf{x}, \int_D x_2^2 f(\mathbf{x}) d\mathbf{x} \right\rangle \\
 &= \left\langle \frac{1024}{6144}, \frac{1024}{3072} \right\rangle = \left\langle \frac{1}{6}, \frac{1}{3} \right\rangle \in \left\langle \left[ \frac{3}{128}, \frac{73}{128} \right], \left[ \frac{25}{256}, \frac{199}{256} \right] \right\rangle
 \end{aligned}$$

and for the expected value of  $X_1 X_2$  is:

$$E[X_1 X_2] = \sum_{B \in \mathcal{H}_\square} \int_B x_1 x_2 f(\mathbf{x}) d\mathbf{x} = \frac{512}{3072} = \frac{1}{6} \in \left[ \frac{3}{128}, \frac{73}{128} \right].$$

The exact values for the covariance matrix of  $\mathbf{Y}$  are:

$$\Sigma(\mathbf{Y}) = \begin{bmatrix} \frac{1}{18} & 0 \\ 0 & \frac{1}{12} \end{bmatrix} \in \begin{bmatrix} \left[ 0, \frac{145}{256} \right] & \left[ -\frac{145}{1024}, \frac{577}{1024} \right] \\ \left[ -\frac{145}{1024}, \frac{577}{1024} \right] & \left[ \frac{231}{4096}, \frac{351}{4096} \right] \end{bmatrix}.$$


---

## 5.2 Conditional Probabilistic Enclosures

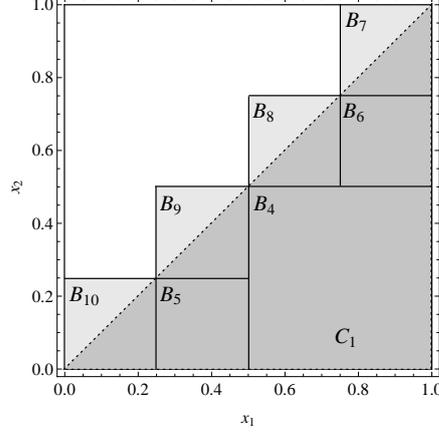
We now address conditional probabilistic features of random vectors (RVs), such as the conditional probability of random vectors as well as the conditional expected values and conditional covariance matrices of random vectors.

**Property 5.4 (Enclosure for the Conditional Probability of a RV)** *Given a PC event  $\mathcal{H}_2$  where  $P(\mathcal{H}_2) > 0$  and any bounded box  $B_{\mathbf{Y}} = I_1 \times \dots \times I_m \in \mathbb{I}\mathbb{R}^m$ , the probability that random vector  $\mathbf{Y}$  takes values in  $B_{\mathbf{Y}}$  given  $\mathcal{H}_2$ ,  $P(\mathbf{Y} \in B_{\mathbf{Y}} | \mathcal{H}_2) = P(Y_1 \in I_1, \dots, Y_m \in I_m | \mathcal{H}_2)$ , is  $P(\mathcal{H}_1 | \mathcal{H}_2)$  where  $\mathcal{H}_1$  is the PC event:*

$$\mathcal{H}_1 = \mathcal{F}(\langle X, D, \{\underline{I}_1 \leq g_1(\mathbf{X}) \leq \overline{I}_1, \dots, \underline{I}_m \leq g_m(\mathbf{X}) \leq \overline{I}_m\} \rangle)$$

*An enclosure for the conditional probability  $P(\mathcal{H}_1 | \mathcal{H}_2)$  can be computed as in property 4.8.*

**Proof.** It follows from definition 3.20, that addresses events induced via random vectors. ■



**Figure 5.4:**  $\mathcal{H}_5$  and its joint box cover  $\mathcal{H}_{\boxplus}$ .

**Example 5.3.** Consider PC *Triangle* =  $\langle\langle X = \langle x_1, x_2 \rangle, D = [0, 1] \times [0, 1], C \rangle, f \rangle$ , where  $f(x_1, x_2) = 2 - 2x_1$  (shown in figure 5.1 (a)),  $C = \{C_1 : x_2 - x_1 \leq 0\}$  (shown in figure 5.4),  $\Omega = D$ , the event  $\mathcal{H}_5 = \mathcal{F}(\langle X, D, \{C_1\} \rangle)$  and its joint box cover  $\mathcal{H}_{\boxplus} = \langle \{B_i : i \in \{4, \dots, 10\}\}, \{B_4, B_5, B_6\} \rangle$  ( $B_i$  are the boxes in figure 5.4). From example 4.6 we have that  $[P](\mathcal{H}_{\boxplus}, f) = [\frac{7}{32}, \frac{17}{32}]$ .

Let  $\mathbf{Y} = \langle X_1, X_2 \rangle$ . Figures 5.2 (a) to (c) show the events  $\mathcal{H}_6$ ,  $\mathcal{H}_7$  and  $\mathcal{H}_8$ , resulting from restricting the values of  $\mathbf{Y}$  to boxes. Table 5.5 shows enclosures for the probabilities of those events intersected with event  $\mathcal{H}_5$  over the boxes  $B_i$ .

$B$	$[P_{\mathcal{H}_5 \cap \mathcal{H}_6}](f, B)$	$[P_{\mathcal{H}_5 \cap \mathcal{H}_7}](f, B)$	$[P_{\mathcal{H}_5 \cap \mathcal{H}_8}](f, B)$	$[P_{\mathcal{H}_5 \cap \mathcal{H}_9}](f, B)$
$B_4$	$[0, 1/4]$		$[0, 1/4]$	$[0, 1/4]$
$B_5$			$[5/64]$	$[5/64]$
$B_6$	$[1/64]$	$[1/64]$		$[1/64]$
$B_7$	$[0, 1/32]$	$[0, 1/32]$		$[0, 1/32]$
$B_8$		$[0, 1/16]$		$[0, 1/16]$
$B_9$			$[0, 3/32]$	$[0, 3/32]$
$B_{10}$				$[0, 1/8]$

**Table 5.5:** Enclosures for the probability of  $B_i \cap \mathcal{H}_j \cap \mathcal{H}_5$ .

Using the values of table 5.5 and property 4.8, enclosures for the conditional probability of events  $\mathcal{H}_6$ ,  $\mathcal{H}_7$  and  $\mathcal{H}_8$  given PC event  $\mathcal{H}_5$  using, respectively, the joint box covers  $\mathcal{H}_{\boxplus \cap 6} = \langle \{B_4, B_6, B_7\}, \{B_6\} \rangle$ ,  $\mathcal{H}_{\boxplus \cap 7} = \langle \{B_6, B_7, B_8\}, \{B_6\} \rangle$  and  $\mathcal{H}_{\boxplus \cap 8} =$

$\langle \{B_4, B_5, B_9\}, \{B_5\} \rangle$  are:

$$\begin{aligned}
 P(\mathcal{H}_6|\mathcal{H}_5) &= \frac{15}{96} \in [P](\mathcal{H}_{\boxplus_6}, \mathcal{H}_{\boxplus_5}, f) = \frac{[P](\mathcal{H}_{\boxplus_6}, f)}{[P](\mathcal{H}_{\boxplus_5}, f)} = \frac{\left[\frac{1}{64}, \frac{19}{64}\right]}{\left[\frac{7}{32}, \frac{17}{32}\right]} = \left[\frac{1}{34}, \frac{19}{14}\right] \\
 P(\mathcal{H}_7|\mathcal{H}_5) &= \frac{1}{8} \in [P](\mathcal{H}_{\boxplus_7}, \mathcal{H}_{\boxplus_5}, f) = \frac{[P](\mathcal{H}_{\boxplus_7}, f)}{[P](\mathcal{H}_{\boxplus_5}, f)} = \frac{\left[\frac{1}{64}, \frac{7}{64}\right]}{\left[\frac{7}{32}, \frac{17}{32}\right]} = \left[\frac{1}{34}, \frac{1}{2}\right] \\
 P(\mathcal{H}_8|\mathcal{H}_5) &= \frac{5}{8} \in [P](\mathcal{H}_{\boxplus_8}, \mathcal{H}_{\boxplus_5}, f) = \frac{[P](\mathcal{H}_{\boxplus_8}, f)}{[P](\mathcal{H}_{\boxplus_5}, f)} = \frac{\left[\frac{5}{64}, \frac{27}{64}\right]}{\left[\frac{7}{32}, \frac{17}{32}\right]} = \left[\frac{5}{34}, \frac{27}{14}\right]
 \end{aligned}$$

Now let  $\mathbf{Y} = \langle X_1 - X_2 \rangle$ . Figure 5.2 (d) shows the event  $\mathcal{H}_9$  resulting from restricting the values of  $\mathbf{Y}$  to a box. Using the values of table 5.5 and property 4.8, an enclosure for the conditional probability of  $\mathcal{H}_9$ , using the joint box cover  $\mathcal{H}_{\boxplus_9} = \langle \{B_i : i \in \{4, \dots, 10\}\}, \{B_5, B_6\} \rangle$ , is:

$$P(\mathcal{H}_9|\mathcal{H}_5) = \frac{7}{8} \in [P](\mathcal{H}_{\boxplus_9}, \mathcal{H}_{\boxplus_5}, f) = \frac{\left[\frac{3}{32}, \frac{21}{32}\right]}{\left[\frac{7}{32}, \frac{17}{32}\right]} = \left[\frac{3}{17}, 3\right]$$

Notice that the example illustrates the computation of conditional probability enclosures using very simple (and crude) joint box covers. As a consequence the obtained enclosures for the conditional probability values provide little information (in some cases the intervals range from almost 0 to values higher than 1).

The exact conditional probability values were obtained analytically.

---

Relying on properties 4.3 and 4.5, an enclosure for the conditional expected value of a random variable given a PC event can be computed as follows.

**Property 5.5 (Enclosure for the Conditional Expected Value of a RV)**

*Given a PC event  $\mathcal{H}$  where  $P(\mathcal{H}) > 0$  and a joint box cover  $\mathcal{H}_{\boxplus}$  of  $\mathcal{H}$ , an enclosure for the conditional expected value of  $\mathbf{Y}$  given  $\mathcal{H}$  is computed as:*

$$[E](\mathbf{Y}, f|\mathcal{H}_{\boxplus}) = \langle [E](Y_1, f|\mathcal{H}_{\boxplus}), \dots, [E](Y_m, f|\mathcal{H}_{\boxplus}) \rangle$$

where,  $\forall_{1 \leq i \leq m}$

$$E[Y_i|\mathcal{H}] \in [E](Y_i, f|\mathcal{H}_{\boxplus}) = \frac{[I](\mathcal{H}_{\boxplus}, g_i f)}{[P](\mathcal{H}_{\boxplus}, f)}$$

**Proof.** Given a random variable  $Y = g(\mathbf{X})$  and a joint PDF  $f$  for  $\mathbf{X}$ , by definition

$$E[Y|\mathcal{H}] = \frac{\int_{\mathcal{H}} g(\mathbf{x})f(\mathbf{x})d\mathbf{x}}{P[\mathcal{H}]}$$

Since, by properties 4.3 and 4.5, respectively,

$$\int_{\mathcal{H}} g(\mathbf{x})f(\mathbf{x})d\mathbf{x} \in [I](\mathcal{H}_{\boxplus}, gf) \quad \text{and} \quad P(\mathcal{H}) \in [P](\mathcal{H}_{\boxplus}, f)$$

then, by the properties of interval arithmetic,

$$E[Y|\mathcal{H}] \in [E](Y, f|\mathcal{H}_{\boxplus}) = \frac{[I](\mathcal{H}_{\boxplus}, gf)}{[P](\mathcal{H}_{\boxplus}, f)}$$

■

The enclosure for the conditional covariance matrix of a random variable given a PC event can be computed using the enclosure for the conditional expected value, as follows.

**Property 5.6 (Enclosure for the Conditional Covariance Matrix)** *Given a PC event  $\mathcal{H}$  where  $P(\mathcal{H}) > 0$  and a joint box cover  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle$  of  $\mathcal{H}$ , an enclosure for the covariance matrix  $\Sigma$  of  $\mathbf{Y}$  is given by:*

$$\begin{aligned} \Sigma_{ij}[\mathbf{Y}|\mathcal{H}] &\in [\Sigma_{ij}](\mathbf{Y}, f|\mathcal{H}_{\boxplus}) = [Cov](Y_i, Y_j, f|\mathcal{H}_{\boxplus}) \\ &= [E](Y_i Y_j, f|\mathcal{H}_{\boxplus}) - [E](Y_i, f|\mathcal{H}_{\boxplus})[E](Y_j, f|\mathcal{H}_{\boxplus}) \end{aligned}$$

*In particular,  $\Sigma_{ii}[\mathbf{Y}|\mathcal{H}] = Var[Y_i|\mathcal{H}] \in [E](Y_i^2, f|\mathcal{H}_{\boxplus}) - [E](Y_i, f|\mathcal{H}_{\boxplus})^2$ .*

**Proof.** Similar to the proof of property 5.3. ■

An example is used to illustrate the previous properties.

**Example 5.4.** Consider again PC *Triangle*, the event  $\mathcal{H}_5 = \mathcal{F}(\langle X, D, \{C_1\} \rangle)$  and its joint box cover  $\mathcal{H}_{\boxplus 5} = \langle \{B_i : i \in \{4, \dots, 10\}\}, \{B_4, B_5, B_6\} \rangle$  ( $B_i$  are the boxes in figure 5.4).

Using the values of tables 5.3 and 5.4<sup>1</sup> we compute:

$$\begin{aligned}
 [I](\mathcal{H}_{\boxplus}, x_1 f) &= \sum_{B \in \mathcal{H}_{\square}} [I_{\mathcal{H}_5}](x_1 f, B) = \left[ \frac{1}{32}, \frac{5}{32} \right] + \left[ \frac{1}{8} \right] = \left[ \frac{5}{32}, \frac{9}{32} \right] \\
 [I](\mathcal{H}_{\boxplus}, x_2 f) &= \left[ \frac{3}{64}, \frac{5}{64} \right] + \left[ \frac{13}{256} \right] = \left[ \frac{25}{256}, \frac{33}{256} \right] \\
 [I](\mathcal{H}_{\boxplus}, x_1^2 f) &= \left[ \frac{3}{256}, \frac{25}{256} \right] + \left[ \frac{81}{1024} \right] = \left[ \frac{93}{1024}, \frac{181}{1024} \right] \\
 [I](\mathcal{H}_{\boxplus}, x_2^2 f) &= \left[ \frac{3}{256}, \frac{25}{256} \right] + \left[ \frac{7}{384} \right] = \left[ \frac{23}{768}, \frac{89}{768} \right] \\
 [I](\mathcal{H}_{\boxplus}, x_1 x_2 f) &= \left[ \frac{3}{256}, \frac{25}{256} \right] + \left[ \frac{25}{768} \right] = \left[ \frac{17}{384}, \frac{50}{384} \right]
 \end{aligned}$$

The exact values for the above integrals, computed analytically, are shown in table 5.6.

$\int_{\mathcal{H}_5} x_1 f(\mathbf{x}) d\mathbf{x}$	$\int_{\mathcal{H}_5} x_2 f(\mathbf{x}) d\mathbf{x}$	$\int_{\mathcal{H}_5} x_1^2 f(\mathbf{x}) d\mathbf{x}$	$\int_{\mathcal{H}_5} x_2^2 f(\mathbf{x}) d\mathbf{x}$	$\int_{\mathcal{H}_5} x_1 x_2 f(\mathbf{x}) d\mathbf{x}$
$\frac{1}{6}$	$\frac{1}{12}$	$\frac{1}{10}$	$\frac{1}{30}$	$\frac{1}{20}$

**Table 5.6:** Exact values for the integrals over the region defined by  $\mathcal{H}_5$ .

From example 4.6 we have that  $[P](\mathcal{H}_{\boxplus}, f) = \left[ \frac{7}{32}, \frac{17}{32} \right]$ , so we can obtain enclosures for the conditional expected values of  $\mathbf{Y}$ ,  $\langle X_1^2, X_2^2 \rangle$  and  $X_1 X_2$  given the PC event  $\mathcal{H}_5$ , as in property 5.5:

$$\begin{aligned}
 [E](\mathbf{Y}, f | \mathcal{H}_{\boxplus}) &= \left\langle \left[ \frac{5}{17}, \frac{9}{7} \right], \left[ \frac{25}{136}, \frac{33}{56} \right] \right\rangle \\
 [E](\langle X_1^2, X_2^2 \rangle, f | \mathcal{H}_{\boxplus}) &= \left\langle \left[ \frac{93}{544}, \frac{181}{224} \right], \left[ \frac{23}{408}, \frac{89}{168} \right] \right\rangle \\
 [E](X_1 X_2, f | \mathcal{H}_{\boxplus}) &= \left[ \frac{1}{12}, \frac{25}{42} \right]
 \end{aligned}$$

Likewise we can obtain enclosures for the conditional covariance matrix of  $\mathbf{Y}$  given the PC event  $\mathcal{H}_5$ , as in property 5.6:

$$[\Sigma](\mathbf{Y}, f | \mathcal{H}_{\boxplus}) = \begin{bmatrix} [Var](X_1, f | \mathcal{H}_{\boxplus}) & [Cov](X_1, X_2, f | \mathcal{H}_{\boxplus}) \\ [Cov](X_2, X_1, f | \mathcal{H}_{\boxplus}) & [Var](X_2, f | \mathcal{H}_{\boxplus}) \end{bmatrix}$$

---

<sup>1</sup>For boxes in  $\mathcal{H}_{\blacksquare}$  the values of table 5.4 are used, by assuming that Taylor models of high order can compute such sharp enclosures.

$$= \begin{bmatrix} [0, 0.7216] & [-0.6744, 0.5412] \\ [-0.6744, 0.5412] & [0, 0.4960] \end{bmatrix}$$

From example 4.3 we have that  $P[\mathcal{H}_5] = \frac{1}{3}$  so, using the values of table 5.6, the exact values for the quantities computed above are:

$$\begin{aligned} E[\mathbf{Y}|\mathcal{H}_5] &= \left\langle \frac{1}{2}, \frac{1}{4} \right\rangle & E[\langle X_1^2, X_2^2 \rangle|\mathcal{H}_5] &= \left\langle \frac{3}{10}, \frac{1}{10} \right\rangle \\ E[X_1 X_2|\mathcal{H}_5] &= \frac{3}{20} & \Sigma(\mathbf{Y}|\mathcal{H}_5) &= \begin{bmatrix} \frac{1}{20} & \frac{1}{40} \\ \frac{1}{40} & \frac{3}{80} \end{bmatrix} \end{aligned}$$

## 5.3 Algorithms

In this section we present algorithms to compute tight enclosures for the probability features of the previous sections.

Computing an enclosure for the probability that a random vector takes values in a box (property 5.1) reduces to compute the probability of the corresponding event. So, algorithm 6 (section 4.6, page 97) is adequate for such computation. Likewise, computing an enclosure for the conditional probability can be done with algorithm 7 (section 4.6, page 102).

Properties 5.2 and 5.3 justify algorithm 10 to compute enclosures for the expected value and covariance matrix of a random vector  $\mathbf{Y}$ . It receives a PC and random vector  $\mathbf{Y}$  and relies on *cReasoning<sub>prob</sub>* function (see property 4.9 for the parametrization) to obtain a partition of  $D$  (line 1) and then, uses such partition to compute and return the required enclosures (line 2).

**Algorithm 10:** *moments*( $\langle\langle X, D, C \rangle, f \rangle, \mathbf{Y}, \varepsilon, \delta$ )

**Input:**  $\langle\langle X, D, C \rangle, f \rangle$ : PC;  $\mathbf{Y} = \langle g_1(\mathbf{X}), \dots, g_m(\mathbf{X}) \rangle$ : random vector;  $\varepsilon, \delta$ : double;

**Output:**  $\langle \mathbf{E}, \Sigma \rangle$ : pair with a tuple of intervals and an array of intervals;

1  $\langle \mathcal{D}_\square, \mathcal{D}_\blacksquare \rangle \leftarrow cReasoning_{prob}(\langle \{D\}, \{D\} \rangle, \{\});$

2 **return**  $\langle [E](\mathbf{Y}, \mathcal{D}_\square, f), [\Sigma](\mathbf{Y}, \mathcal{D}_\square, f) \rangle;$

The correctness of the algorithm is guaranteed since the initial pair  $\langle\{D\}, \{D\}\rangle$  is a joint box cover of  $\mathcal{D} = \mathcal{F}(\langle X, D, \{\}\rangle)$  and, inside  $cReasoning_{prob}$ , every iteration of  $crStep$  produces tighter joint box covers of  $\mathcal{D}$ . So, when  $cReasoning$  stops  $\mathcal{D}_{\boxplus}$  is a joint box cover of  $\mathcal{D}$  and, by properties 5.2 and 5.3,  $\forall Y_i \in \mathbf{Y} \ E[Y_i] \in [E](Y_i, \mathcal{D}_{\boxplus}, f)$  and  $\forall Y_i \in \mathbf{Y} \forall Y_j \in \mathbf{Y} \ Cov(Y_i, Y_j) \in [Cov](Y_i, Y_j, \mathcal{D}_{\boxplus}, f)$ .

Algorithm 11 computes enclosures for the conditional expected value (as in property 5.5) and covariance matrix (as in property 5.6) of a random vector  $\mathbf{Y}$ , given the conditioning event. It receives a PC where the constraints are those associated with the conditioning event  $\mathcal{H}$  and random vector  $\mathbf{Y}$ . It relies on function  $cReasoning_{prob}$  to obtain successively tighter covers of  $\mathcal{H}$ , until the required probability accuracy  $\delta$  is achieved (line 1), and then uses the resulting cover to compute and return the required enclosures (line 2).

**Algorithm 11:**  $conditionalMoments(\langle\langle X, D, C_{\mathcal{H}}\rangle, f\rangle, \mathbf{Y}, \varepsilon, \delta)$

**Input:**  $\langle\langle X, D, C_{\mathcal{H}}\rangle, f\rangle$ : PC;  $\mathbf{Y} = \langle g_1(\mathbf{X}), \dots, g_m(\mathbf{X})\rangle$ : random vector;  $\varepsilon, \delta$ : double;

**Output:**  $\{\mathbf{E}_{\mathcal{H}}, \mathbf{\Sigma}_{\mathcal{H}}\}$ : pair with a tuple of intervals and an array of intervals;

1  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle \leftarrow cReasoning_{prob}(\langle\{D\}, \emptyset\rangle, C_{\mathcal{H}})$ ;

2 **return**  $\langle [E](\mathbf{Y}, f|\mathcal{H}_{\boxplus}), [\Sigma](\mathbf{Y}, f|\mathcal{H}_{\boxplus}) \rangle$ ;

Using a similar reasoning as in algorithm 10, the correctness of algorithm 11 is guaranteed .

On both algorithms, the accuracy of the enclosure for the total probability defines the stopping criterion, since this quantity is a good indicator for the accuracy of the computed enclosures. Likewise, the probability enclosures of the computed boxes define the order criterion of such algorithms.

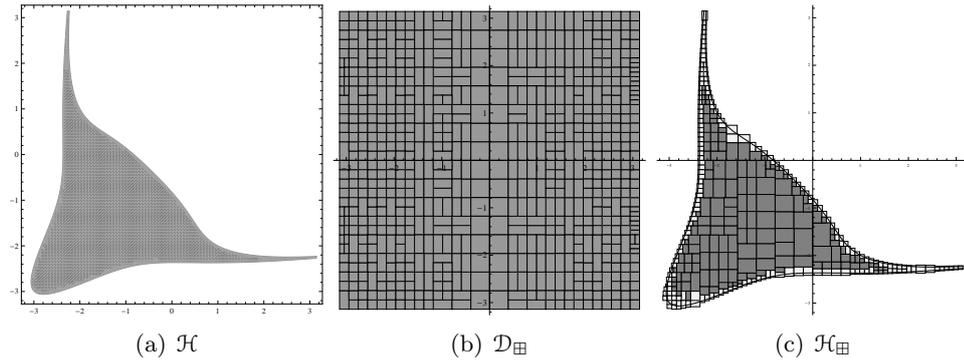
**Example 5.5.** Consider PC  $\langle\langle X, D, C\rangle, f\rangle$  where  $X = \langle x, y\rangle$ ,  $D = [-\pi, \pi] \times [-\pi, \pi]$ ,  $C_{\mathcal{H}} = \{-1.5 \leq e^{x+1.5}(y+1.5)^3 + e^{y+1.5}(x+1.5)^3 \leq 8\}$ ,  $f = \frac{1+\cos(x^2+y)}{4\pi^2}$  is a custom distribution defined over  $\Omega = D$  (figure 5.6 (a)) and  $\mathcal{H}$  is a PC event (shown in figure 5.5(a)).

Figures 5.5 (b) and (c) present the covers resulting from applying, respectively, algorithms 10 and 11 to compute enclosures for the unconditional and conditional (given event  $\mathcal{H}$ ) expected values and covariance matrices of  $\mathbf{Y} = \langle X_1, X_2 \rangle$  with  $\delta = 0.1$ . The

results were:

$$\begin{aligned}
 [E](\mathbf{Y}, \mathcal{D}_{\boxplus}, f) &= \langle [-0.1337, 0.1331], [-0.3753, -0.1271] \rangle \\
 [\Sigma](\mathbf{Y}, \mathcal{D}_{\boxplus}, f) &= \begin{bmatrix} [2.8001, 3.8311] & [-0.3329, 0.3302] \\ [-0.3329, 0.3302] & [2.4031, 3.3819] \end{bmatrix} \\
 [E](\mathbf{Y}, f|\mathcal{H}_{\boxplus}) &= \langle [-1.9294, -0.6117], [-2.0134, -0.7278] \rangle \\
 [\Sigma](\mathbf{Y}, f|\mathcal{H}_{\boxplus}) &= \begin{bmatrix} [0, 3.5560] & [-3.3417, 1.7674] \\ [-3.3417, 1.7674] & [0, 3.5664] \end{bmatrix}
 \end{aligned}$$

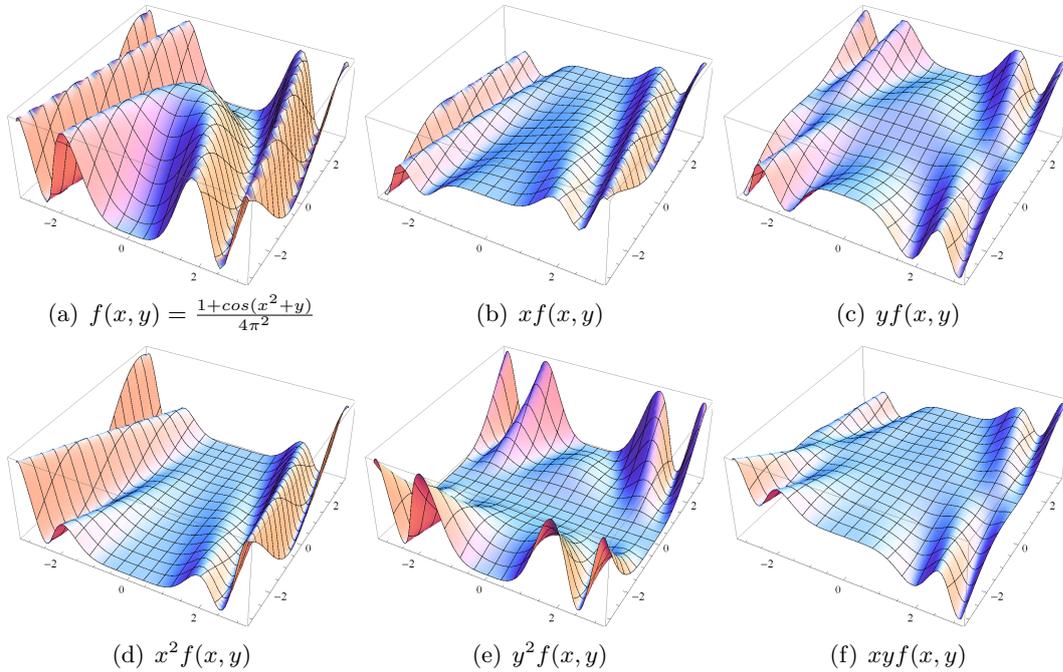
Figure 5.6 shows the integrand functions involved in the previous computations.



**Figure 5.5:** Event  $\mathcal{H}$  and joint box covers of  $D$  and  $\mathcal{H}$  computed with algorithms 10 and 11 with a safe integration method. The boundary boxes are light gray and the inner boxes are gray.

## 5.4 Probability Distributions

Probability distributions describe the probability that random variables fall within a specified range of values. There are two main types of models for probability distributions: parametric, which are based on parametric families of probability distributions (e.g. Gaussian, uniform) and nonparametric, whenever they do not follow any standard probability distribution. For nonparametric models, techniques are used to compute the distribution of a random variable without assuming a particular shape for it.



**Figure 5.6:** Integrand functions used to computed the expected values and variances.

In this section an algorithm is presented to compute a nonparametric conditional probability distribution of a random vector and output it in tabular form, for subsequent processing. This information is necessary in many real world applications as shown in part III of this thesis.

Additionally, since a graphical output is an important means to convey information, the result of the algorithm can be used for graphical plotting, thus providing an adequate display of the shape of the distribution of the variables of interest across their range. Its purpose is to graphically summarize the distribution and to illustrate: 1) the location of the values, 2) its scale, 3) its skewness, and 4) whether multiple modes in the distribution exist.

In our context, given a PC  $\langle\langle X, D, C \rangle, f\rangle$ , the aim is to compute the probability distribution of (a subset of) the implicit random vector  $\mathbf{X}$  given a PC event  $\mathcal{H}$ . For this purpose, some kind of discretization is convenient, so the range of the random variables of interest is divided into a grid of equal-sized intervals.

**Definition 5.1 (m-Dimensional Grid on S)** Given a tuple of positive values  $\alpha = \langle \alpha_1, \dots, \alpha_m \rangle$ , a grid  $\mathcal{G}_\alpha$  on  $S \subseteq \mathbb{R}^m$  is a set of almost disjoint grid boxes such that:

$$\forall G = I_1 \times \dots \times I_m \in \mathcal{G}_{\langle \alpha_1, \dots, \alpha_m \rangle} \quad \forall 1 \leq i \leq m \quad \exists k \in \mathbb{Z} \quad I_i = [k\alpha_i, (k+1)\alpha_i]$$

Each  $\alpha_i$  defines the spacing between two consecutive grid points on the  $i^{\text{th}}$  dimension.  $S$  is an  $m$ -dimensional box that defines the grid bounds.

The conditional probability for each grid box is obtained by computing the probability of the conjunction of this box and the conditioning event. For this purpose we compute an adequate joint box cover of the conditioning event.

**Definition 5.2 (Grid Joint Box Cover)** A joint box cover  $\mathcal{H}_\boxplus$  is a grid joint box cover wrt a grid  $\mathcal{G}_\alpha$  iff every box in the cover is included in a single grid box of  $\mathcal{G}_\alpha$ .

Algorithm 12 computes the conditional probability distribution of a random vector  $\mathbf{Z} = \langle X_{i_1}, \dots, X_{i_m} \rangle$  given a PC event  $\mathcal{H}$ . It outputs an  $m$ -dimensional array  $M$  of probability enclosures and a box  $H$  that encloses the conditioning event  $\mathcal{H}$ , defining the region characterized by the distribution (line 16).

The algorithm is composed of four main steps:

1. Compute a grid  $\mathcal{G}_\alpha$  on  $H$  (lines 1 – 5). The grid bounding box  $H$  is computed by reducing the initial search space  $D$  through constraint reasoning (*cReasoning* is used on the constraints  $C_{\mathcal{H}}$ , during a predefined amount of time imposed by *stop<sub>T</sub>*) and making the union hull of the boxes in the resulting outer box cover (lines 1 – 2). The grid spacings  $\alpha$  are based on  $H$  and on the number of partitions for each dimension  $\langle l_1, \dots, l_m \rangle$  (line 3). To ensure that  $H$  bounds are on the grid,  $H$  is adequately inflated<sup>1</sup> (lines 4 – 5). Time spent in this step should be very

<sup>1</sup>The implementation of such operation imposes that inflating does not go beyond the limits of  $\Pi_{Z_{idx}}(D)$ .

**Algorithm 12:**  $\text{marginalDistribution}(\langle\langle X, D, C \rangle, f \rangle, C_{\mathcal{H}}, Z_{idx}, \varepsilon, \delta, L)$ 

**Input:**  $\langle\langle X, D, C \rangle, f \rangle$ : PC;  $C_{\mathcal{H}}$ : set of constraints;  $Z_{idx} = \langle i_1, \dots, i_m \rangle$ : tuple of variables indexes;  $\varepsilon, \delta$ : double;  $L = \langle l_1, \dots, l_m \rangle$ : tuple of integers  
**Output:**  $\langle M, H \rangle$ : pair with an  $m$ -dimensional matrix of intervals and an  $m$ -box;

- 1  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle \leftarrow cReasoning(\langle\{D\}, \emptyset \rangle, C_{\mathcal{H}}, split_2, inner_d, eligible_{\varepsilon}, order_W, stop_T)$ ;
- 2  $\forall_{1 \leq i \leq n} H_i \leftarrow \biguplus_{B \in \mathcal{H}_{\square}} \Pi_i(B)$ ;
- 3  $\forall_{1 \leq j \leq m} \alpha_j \leftarrow wid(H_{i_j})/l_j$ ;
- 4  $\forall_{1 \leq j \leq m} (I_j \leftarrow \lceil floor(inf(H_{i_j})/\alpha_j), ceil(sup(H_{i_j})/\alpha_j) \rceil); \quad H_{i_j} \leftarrow \alpha_j I_j$ ;
- 5  $H_D \leftarrow H_1 \times \dots \times H_n; \quad H \leftarrow \Pi_{Z_{idx}}(H_D)$ ;
- 6  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle \leftarrow cReasoning(\langle\{H_D\}, \emptyset \rangle, C_{\mathcal{H}}, split_{\alpha}, inner_d, eligible_{\alpha}, order_{\downarrow}, false)$ ;
- 7  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle \leftarrow cReasoning(\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle, C_{\mathcal{H}}, split_2, inner_d, eligible_{\varepsilon}, order_P, stop_{\delta})$ ;
- 8  $\forall_{1 \leq k_1 \leq wid(I_1)} \dots \forall_{1 \leq k_m \leq wid(I_m)} M[k_1] \dots [k_m] \leftarrow [0]; \quad P \leftarrow [0]$ ;
- 9 **foreach** ( $B \in \mathcal{H}_{\square}$ ) **do**
- 10      $\forall_{1 \leq j \leq m} B_j \leftarrow \Pi_{\langle i_j \rangle}(B)$ ;
- 11      $\forall_{1 \leq j \leq m} k_j \leftarrow ceil(sup(B_j)/\alpha_j) - inf(I_j)$ ;
- 12      $M[k_1] \dots [k_m] \leftarrow M[k_1] \dots [k_m] + [P_{\mathcal{H}}](B)$ ;
- 13      $P \leftarrow P + [P_{\mathcal{H}}](B)$ ;
- 14 **end**
- 15  $\forall_{1 \leq k_1 \leq wid(I_1)} \dots \forall_{1 \leq k_m \leq wid(I_m)} M[k_1] \dots [k_m] \leftarrow \frac{M[k_1] \dots [k_m]}{P}$ ;
- 16 **return**  $\langle M, H \rangle$ ;

small and the computed boxes are discarded, although  $H_D$  is given as input to the next step.

2. Compute a grid joint box cover  $\mathcal{H}_{\boxplus}$  of  $\mathcal{H}$  wrt  $\mathcal{G}_{\alpha}$  (line 6). Function  $cReasoning$  is used over  $H_D$  with a grid oriented parametrization, i.e.,  $split_{\alpha}$  splits the boxes on grid points and  $eligible_{\alpha}$  chooses boxes that are not completely inside a grid box. The  $stop \equiv false$  implies that the algorithm only stops when there are no more eligible boxes (see appendix A for details on  $split_{\alpha}$  and  $eligible_{\alpha}$ ).
3. Refine the grid joint box cover with  $cReasoning$  until the required accuracy for the total probability enclosure is achieved (imposed by  $stop_{\delta}$ ) or every box is already sufficiently small (imposed by  $eligible_{\varepsilon}$ ) (line 7).
4. Calculate the probability distribution for  $\mathbf{Z}$ . For each grid dimension on  $H$ , the

number of grid units (stored in  $I_s$ ) defines the number of cells in the corresponding array dimension, which are initialized to zero (line 8). For each box in the outer cover, its corresponding array cell is identified (lines 10 – 11) and its probability contribution is added up to the value in that cell (line 12). The total probability of the conditioning event is computed during this process (lines 8 and 13) and used in the end to normalize the computed probabilities (line 15).

In practice, when computing marginal conditional distributions for each  $X_i$  and for combinations of such random variables, algorithm 12 does not need to be called several times (one for each intended marginal distribution). In fact, once computed the joint box cover of event  $\mathcal{H}$ , it can be used to compute any set of marginal distributions.

In some situations it is desirable to directly provide the grid spacings,  $\alpha$ , instead of computing them from the number of partitions for the grid. This is easily accomplished by skipping step 1 above (and replace  $H_D$  by  $D$  in line 6). Likewise, in some cases, it is unnecessary, or even undesirable, to normalize the computed distribution, thus skipping lines 13 and 15.

**Example 5.6.** Consider PC  $\langle\langle X, D, C \rangle, f\rangle$  where  $X = \langle x, y \rangle$ ,  $D = [-\pi, \pi] \times [-\pi, \pi]$ ,  $f = \frac{e^{-\frac{1}{2}(x^2+y^2)}}{2\pi}$  is a bi-normal distribution defined over  $\Omega = \mathbb{R}^n$ . Consider  $C_{\mathcal{H}} = \{x^2y + y^2x \leq 0.5\} \subseteq C$  and the corresponding PC event  $\mathcal{H}$ .

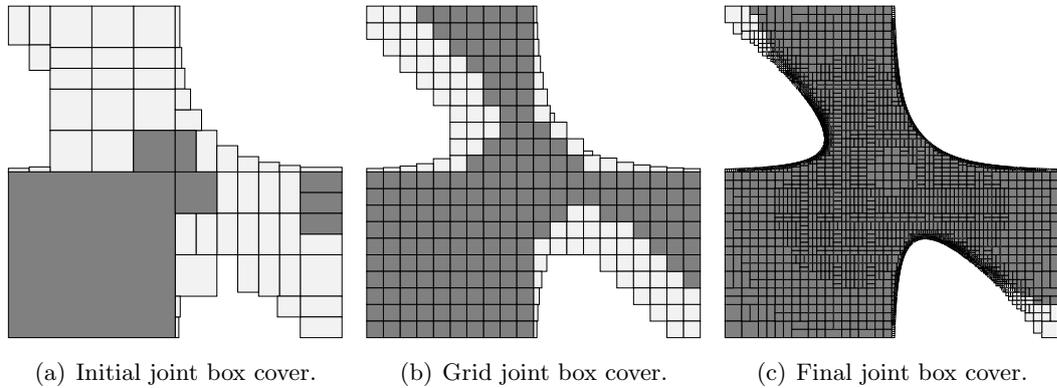
Figure 5.7 presents the joint box covers for event  $\mathcal{H}$ , obtained by algorithm 12 when applied to the above PC,  $C_{\mathcal{H}}$  and  $\mathbf{Z} = \langle X, Y \rangle$ : a) shows the result of step 1 for  $L = \langle 20, 20 \rangle$  and time = 30ms; b) presents the result of step 2 for  $\langle \alpha_1, \alpha_2 \rangle = \langle 0.314159, 0.314159 \rangle$ ; and c) shows result of step 3 with  $\delta = 0.01$  and  $\varepsilon = 10^{-15}$ .

Figures 5.8 (a) and (b) present the conditional probability distributions of, respectively,  $\langle X, Y \rangle | \mathcal{H}$  and  $X | \mathcal{H}$ , computed by algorithm 12 from the final joint box cover shown in figure 5.7 (c). In figure 5.8 (a), darker colors represent more likely regions.

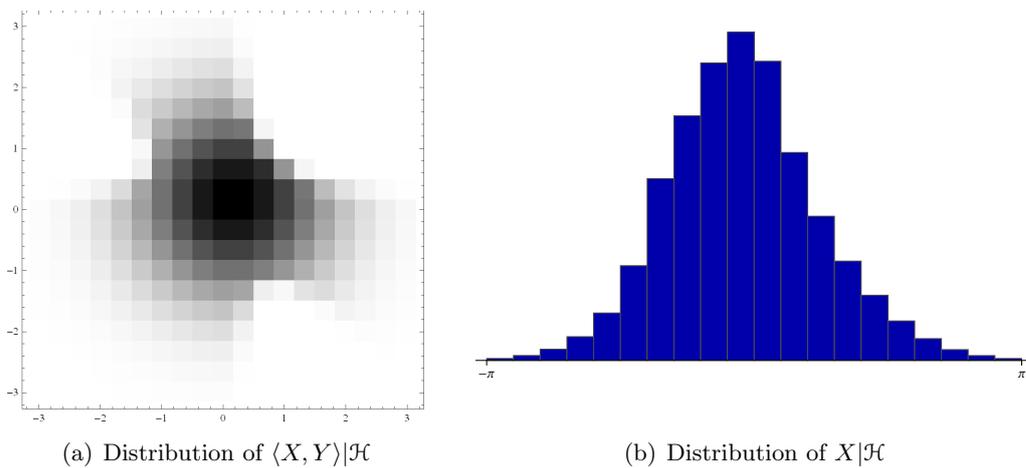
---

#### 5.4.1 Probability Distributions for Functions of Random Vectors

Given a probabilistic constraint space, we are also interested in the more general case of computing nonparametric probability distributions of  $\mathbf{Y} = \langle Y_1, \dots, Y_m \rangle$  where each



**Figure 5.7:** Joint box covers produced when computing probability distributions.



**Figure 5.8:** Marginal probability distributions.

$Y_i = g_i(\mathbf{X})$  is as a function of  $\mathbf{X}^1$  defined on the same probability space.

The approach to compute such probability distributions is very similar, in concept, to the approach presented in the previous section, since a grid over the range of  $\mathbf{Y}$  is considered. Nevertheless, in the present case, two distinct spaces exist: (a) the  $n$ -dimensional box  $D$ , over which the conditioning event  $\mathcal{H}$  is defined, and (b) the  $m$ -dimensional box  $D_Y$  which is the range of  $\mathbf{Y}$  over  $D$ . In fact we deal with  $n + m$ -dimensional *virtual* boxes  $B \times [g_1](B) \times \cdots \times [g_m](B)$ .

Space  $D$  is subject to constraint reasoning to obtain joint box covers of the conditioning

<sup>1</sup>See property 3.10.

event. Moreover space  $D_Y$  is computed from space  $D$  using the corresponding inclusion functions,  $[g_i]$ , and every joint box cover in space  $D$  has a corresponding set of boxes in space  $D_Y$  (that is not necessarily almost disjoint).

Hence there is no direct manipulation of space  $D_Y$  through constraint reasoning and no grid will be imposed, directly, over such space. This is problematic since it may bias the computation of probability distributions. Therefore this effect is minimized by choosing boxes of the joint box cover in space  $D$  whose corresponding box in  $D_Y$  is not completely inside a grid box and, indirectly, obtain a set of boxes in space  $D_Y$ , where each box is contained in a single grid box of  $D_Y$  or is *small enough* to have a negligible impact in the computed distribution.

Figure 5.9 illustrates the correspondence between boxes in space  $D$  and boxes in space  $D_Y$  (in this case 1-dimensional boxes, i.e., intervals), for three boxes of a joint box cover of event  $\mathcal{H}$ . The random vector  $\mathbf{Y}$  is defined as  $\mathbf{Y} = \langle g[\mathbf{X}] = X_1 + X_2 \rangle$ . Considering  $\alpha = 0.5$ , only box  $B_2$  in space  $D$  originates a box in space  $D_Y$  that is inside a grid box. The others originate boxes that span for more that one grid box.

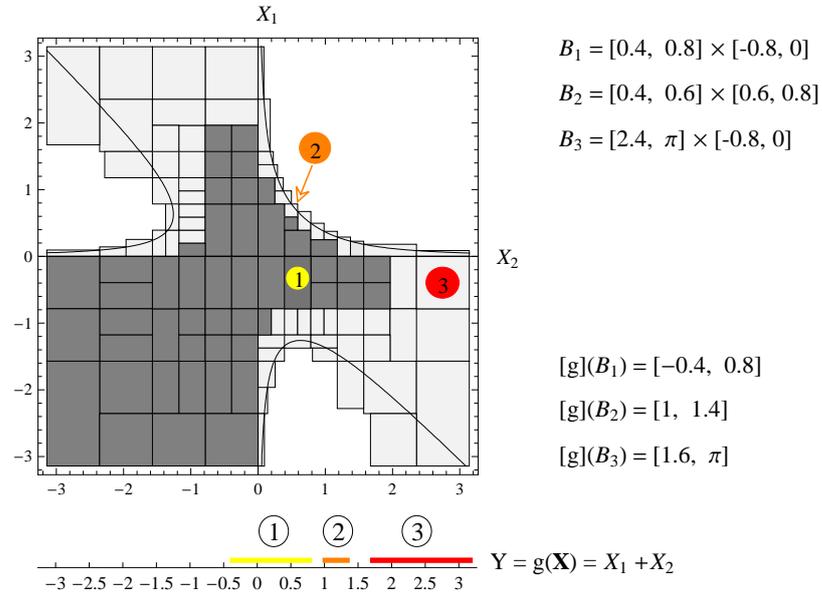


Figure 5.9: Joint box cover of  $\mathcal{H}$  (top) and correspondent joint box cover of  $\mathbf{Y}$  (bottom).

**Algorithm 13:** *marginalDistributionY*( $\langle\langle X, D, C \rangle, f \rangle, C_{\mathcal{H}}, G, \varepsilon, \delta, L$ )

**Input:**  $\langle\langle X, D, C \rangle, f \rangle$ : PC;  $C_{\mathcal{H}}$ : set of constraints;  $G = \{[g_1], \dots, [g_m]\}$ : set of inclusion functions;  $\varepsilon, \delta$ : double;  $L = \langle l_1, \dots, l_m \rangle$ : tuple of integers  
**Output:**  $\langle M, H_Y \rangle$ : pair with an  $m$ -dimensional array of intervals and a box;

- 1  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle \leftarrow cReasoning(\langle\{D\}, \emptyset \rangle, C_{\mathcal{H}}, split_2, inner_d, eligible_{\varepsilon}, order_W, stop_T)$ ;
- 2  $\forall_{1 \leq i \leq m} H_{Y_i} \leftarrow \bigcup_{B \in \mathcal{H}_{\square}} [g_i](B)$ ;  $\forall_{1 \leq i \leq n} H_i \leftarrow \bigcup_{B \in \mathcal{H}_{\square}} \Pi_i(B)$ ;
- 3  $\forall_{1 \leq i \leq m} \alpha_i \leftarrow wid(H_{Y_i})/l_i$ ;
- 4  $\forall_{1 \leq i \leq m} I_i \leftarrow [floor(inf(H_{Y_i})/\alpha_i), ceil(sup(H_{Y_i})/\alpha_i)]$ ;
- 5  $H_Y \leftarrow \alpha_1 I_1 \times \dots \times \alpha_m I_m$ ;  $H \leftarrow H_1 \times \dots \times H_n$ ;
- 6  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle \leftarrow cReasoning(\langle\{H\}, \emptyset \rangle, C_{\mathcal{H}}, split_2, inner_{\alpha}, eligible_{\varepsilon}, order_P, stop_{\delta})$ ;
- 7  $\forall_{1 \leq k_1 \leq wid(I_1)} \dots \forall_{1 \leq k_m \leq wid(I_m)} M[k_1] \dots [k_m] \leftarrow [0]$ ;  $P \leftarrow [0]$ ;
- 8 **foreach** ( $B \in \mathcal{H}_{\square}$ ) **do**
- 9      $\forall_{1 \leq i \leq m} B_i \leftarrow [g_i](B)$ ;
- 10      $\forall_{1 \leq i \leq m} J_i \leftarrow [(floor(inf(B_i)/\alpha_i) - inf(I_i)) + 1, ceil(sup(B_i)/\alpha_i) - inf(I_i)]$ ;
- 11      $indexSet = \{\langle j_1, \dots, j_m \rangle : \forall_{1 \leq i \leq m} j_i \in J_i \wedge j_i \in \mathbb{N}\}$ ;
- 12     **foreach** ( $\langle j_1, \dots, j_m \rangle \in indexSet$ ) **do**
- 13          $M[j_1] \dots [j_m] \leftarrow M[j_1] \dots [j_m] + [P_{\mathcal{H}}](B)$ ;
- 14     **end**
- 15     **if** ( $inner_d(B, C_{\mathcal{H}})$ ) **then**  $P \leftarrow P + [P](B)$  **else**  $P \leftarrow P + [P_{\mathcal{H}}](B)$ ;
- 16 **end**
- 17  $\forall_{1 \leq k_1 \leq wid(I_1)} \dots \forall_{1 \leq k_m \leq wid(I_m)} M[k_1] \dots [k_m] \leftarrow \frac{M[k_1] \dots [k_m]}{P}$ ;
- 18 **return**  $\langle M, H_Y \rangle$ ;

To compute the probability distribution of a random vector  $\mathbf{Y}$ , an algorithm similar to algorithm 12 is developed, with the necessary adaptations to incorporate the differences. Besides replacing parameter  $Z_{idx}$  by a set of inclusion functions (one for each  $Y_j$ ), the four steps are adapted as follows:

1. The grid  $\mathcal{G}_{\alpha}$  is now computed on space  $D_Y$ . So, the grid bounding box  $H_Y = [g_1](H) \times \dots \times [g_m](H)$  is computed from the boxes in the outer box cover of the first call to *cReasoning*, by applying the corresponding inclusion functions to each box and then considering their union hull (lines 1 – 5).
2. Since there is no point in computing a grid joint box cover of  $\mathcal{H}$  (the grid will not be in space  $D$ ), this step is discarded.
3. This step computes a grid joint box cover with *cReasoning* until the required accuracy for the total probability enclosure is achieved (imposed by  $stop_{\delta}$ ) or every

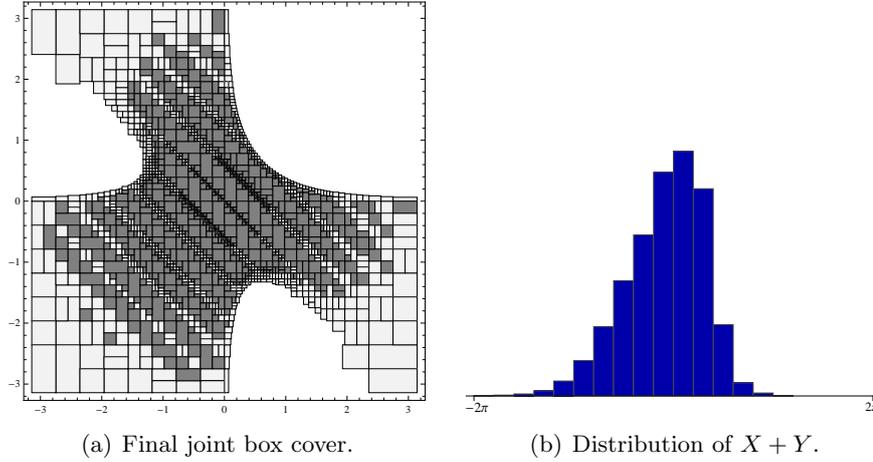
box is already sufficiently small (imposed by  $eligible_\varepsilon$ ) (line 7). The boxes are considered inner boxes if they satisfy the constraints  $C_{\mathcal{H}}$  and their corresponding box in space  $D_Y$  is inside a single grid box (imposed by the  $inner_\alpha$  predicate described in appendix A). This extra condition is important as it makes such boxes good candidates for processing (by assigning them a probability enclosure with a large width) and, eventually, transform their descendants into inner boxes or *small enough* such that their contribution is negligible and does not bias the probability distribution.

4. This step calculates the probability distribution for  $\mathbf{Y}$ . The main difference is that each relevant box in  $D_Y$  (computed on line 9) can span for more than one grid box. So, more than one corresponding array cell can be identified (lines 10 – 11) and the box probability contribution is added up to the values in those cells (lines 12 – 14). Notice that, when computing the total probability of the conditioning event (line 15), boxes are considered inner boxes when they satisfy the  $inner_d$  predicate (wrt the constraints  $C_{\mathcal{H}}$ ) and the corresponding probability enclosure is added up.

**Example 5.7.** Consider PC and data of example 5.6. Figure 5.10 presents the result of algorithm 13 when applied to the referred PC,  $C_{\mathcal{H}}$  and  $\mathbf{G} = \{[g](\langle X, Y \rangle) = X + Y\}$ ,  $L = \langle 20 \rangle$ , time = 30ms,  $\delta = 0.2$  and  $\varepsilon = 10^{-15}$ : a) shows the final joint box cover of event  $\mathcal{H}$ ; b) shows the obtained distribution of  $Y_1 = X + Y$ .

---

Marginal probability distributions of a random vector that mixes random variables of  $\mathbf{X}$  with functions over that random vector (i.e.,  $g_1[\mathbf{X}], \dots, g_n[\mathbf{X}]$ ) can also be computed by creating a more generic algorithm that combines algorithms 12 and 13. In this case special care must be taken when classifying inner boxes after line 6 of algorithm 12. Before the next call to  $cReasoning$ ,  $\mathcal{H}_\blacksquare$  must be recomputed to include exclusively those boxes that satisfy the predicate  $inner_\alpha$ .



**Figure 5.10:** Joint box cover and distribution produced by algorithm 13.

## 5.5 Alternative Approximate Computations

Instead of using the safe integration method provided by the Taylor model based integration, Monte Carlo integration can be used as an alternative approach to compute approximate enclosures for unconditional or conditional expected values and variances and approximate probability distributions.

From the generic definition 4.9, an approximate enclosure  $\widehat{[I]}(\mathcal{H}_{\boxplus}, h)$  for  $[I](\mathcal{H}_{\boxplus}, h)$  in property 4.3 can be achieved by replacing  $[I_{\mathcal{Y}_t}](h, B)$  by its approximate Monte Carlo enclosure  $\widehat{[I_{\mathcal{Y}_t}]}(h, B)$ .

Using such approximate enclosure,  $\widehat{[I]}(\mathcal{H}_{\boxplus}, \cdot)$ , in property 5.2 and, subsequently in property 5.3, originates the corresponding approximate Monte Carlo enclosures for the expected value,  $\widehat{[E]}(\mathbf{Y}, \mathcal{H}_{\boxplus}, \cdot)$ , and covariance matrix,  $\widehat{[\Sigma]}(\mathbf{Y}, \mathcal{H}_{\boxplus}, \cdot)$ , of a random vector.

Likewise, replacing  $[I](\mathcal{H}_{\boxplus}, \cdot)$  and  $[P](\mathcal{H}_{\boxplus}, \cdot)$  by their approximate counterparts in property 5.5 and, subsequently, in property 5.6, we obtain the approximate Monte Carlo enclosures for the conditional expected value  $\widehat{[E]}[\mathbf{Y}, \cdot | \mathcal{H}_{\boxplus}]$  and for the conditional covariance matrix,  $\widehat{[\Sigma]}[\mathbf{Y}, \cdot | \mathcal{H}_{\boxplus}]$ .

In algorithms 10, 11, 13 and 13 (and its auxiliary functions) both in the pseudo code and in the *cReasoning* parametrization, the considered enclosures can be replaced by the corresponding approximate Monte Carlo enclosures trading computational speed with a potential loss of correctness.

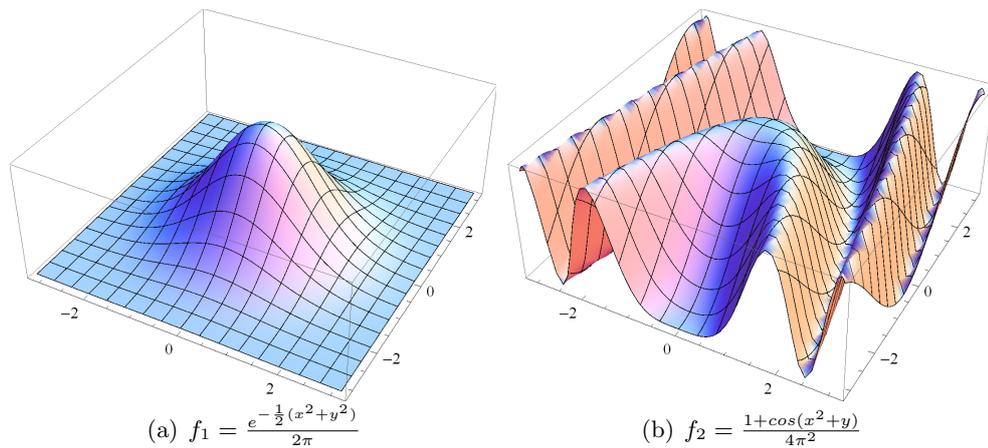
## 5.6 Experimental Results

In this section we present the results of applying the algorithms proposed in the previous section to a set of events. Two versions of the algorithms, implemented over RealPaver 1.0, are considered in the experiments: PCTM uses the validated quadrature method based on Taylor models and provides safe enclosures of the computed quantities; PCMC uses Monte Carlo integration and provides estimates for the computed quantities. All the experiments were performed on an Intel Core Duo at 1.83 GHz with 1 GB of RAM.

### 5.6.1 The PC Events

Consider the CCSP  $CP$  where  $X = \langle x, y \rangle$ ,  $D = [-\pi, \pi] \times [-\pi, \pi]$  and  $C = \{C_i : i \in \{1, 6, 7, 11, \dots, 18\}\}^1$ , compactly defined in Table 5.7. Consider two PCs,  $\langle CP, f_1 \rangle$  and  $\langle CP, f_2 \rangle$ , where  $f_1$  is the bi-normal distribution (figure 5.11 (a)) defined over  $\Omega_1 = \mathbb{R}^2$  and  $f_2$  is a custom distribution defined over  $\Omega_2 = D$  (figure 5.11 (b)). Consider the events  $\mathcal{H}_i$  (shown in figure 5.12), whose associated subset of constraints is defined in the last column of table 5.7.

We can observe that PDF  $f_1$  has axial symmetry wrt axis  $z$ , while  $f_2$  has not. Moreover events  $\mathcal{H}_1$ ,  $\mathcal{H}_5$  and  $\mathcal{H}_7$  have reflection symmetry:  $\mathcal{H}_1$  wrt to the diagonal axis  $x = y$ ;  $\mathcal{H}_5$  wrt to axis  $x$ ,  $y$  and the diagonal axis  $x = y$  and  $x = -y$ ; and  $\mathcal{H}_7$  wrt to axis  $y$ .



**Figure 5.11:** Probability density functions used for testing.

<sup>1</sup>The event indexes were maintained from chapter 4.

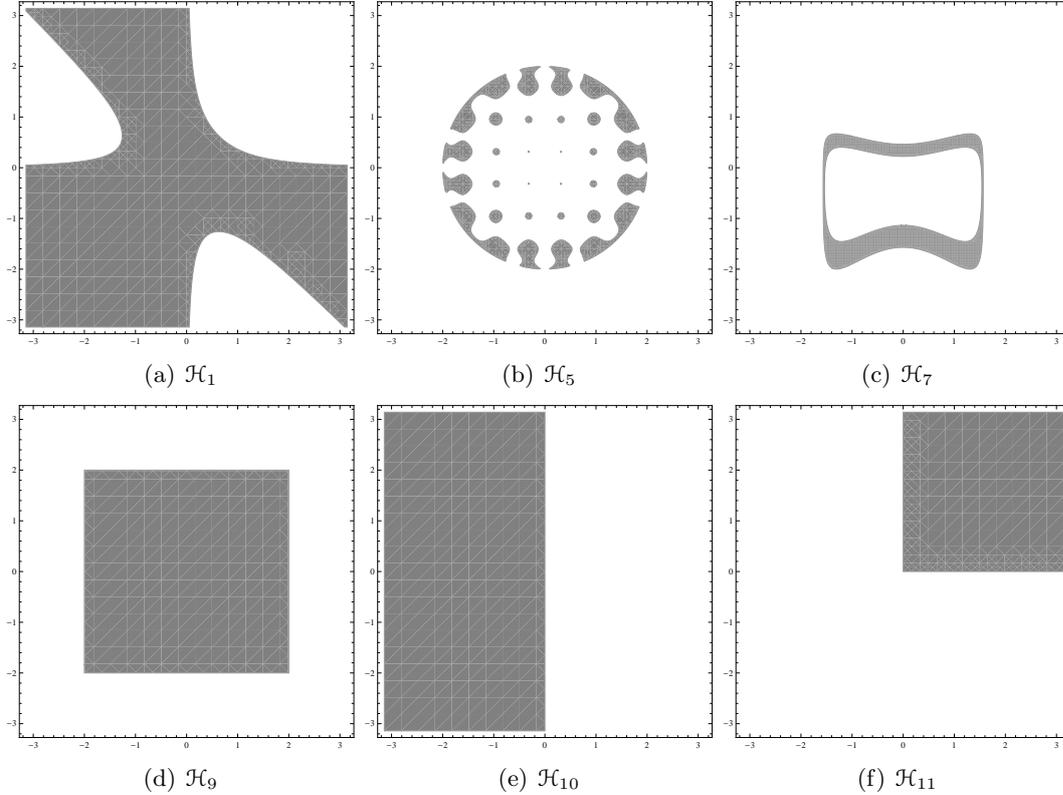


Figure 5.12: Events used for testing.

### 5.6.2 Probabilistic Enclosures

Tables 5.8 and 5.9 show the results of applying both versions of algorithm 6 (in chapter 4) to the events  $\mathcal{H}_9$ ,  $\mathcal{H}_{10}$  and  $\mathcal{H}_{11}$  resulting from restricting the values of random variable  $X$  and  $Y$ , respectively, to the boxes  $[-2, 2] \times [-2, 2]$ ,  $[-\pi, 0] \times [-\pi, \pi]$  and  $[0, \pi] \times [0, \pi]$ , with PDF  $f_1$  (table 5.8) and PDF  $f_2$  (table 5.9). The PCTM version of the algorithm was parameterized with  $\varepsilon = 10^{-15}$ ,  $\delta = 0.001$  and a Taylor model order of 2 and the PCMC version with  $\varepsilon = 10^{-15}$ ,  $\delta = 0.005$  and  $N = 100$  random sample points.

The first two columns refer to the PCTM version results: the enclosure for the probability of the event,  $[P(\mathcal{H})]$ , and the CPU time, in milliseconds, to compute it. The last three columns refer to the PCMC version results: the numerical approximate probability of the event,  $\widehat{P}(\mathcal{H})$ , the CPU time, in milliseconds, to compute it and a lower bound  $\mathcal{E}$  for the relative error of  $\widehat{P}(\mathcal{H})$  based on the safe enclosure  $[P(\mathcal{H})]$ , as in formula 4.9.

$C_1$	$x^2y + y^2x \leq 0.5$	$C_{\mathcal{H}_1} = \{C_1\}$
$C_6, C_7$	$10\cos(5x)^2 + 10\cos(5y)^2 \leq (x^2 + y^2)^2, x^2 + y^2 \leq 4$	$C_{\mathcal{H}_5} = \{C_6, C_7\}$
$C_{10}, C_{11}$	$1 \leq \cos(x)(e^{\cos(y)} - \cos(x) - \sin(y))^5 \leq 6$	$C_{\mathcal{H}_7} = \{C_{10}, C_{11}\}$
$C_{12}, \dots, C_{15}$	$-2 \leq x \leq 2, -2 \leq y \leq 2$	$C_{\mathcal{H}_9} = \{C_{12}, \dots, C_{15}\}$
$C_{16}$	$x \leq 0$	$C_{\mathcal{H}_{10}} = \{C_{16}\}$
$C_{17}, C_{18}$	$0 \leq x, 0 \leq y$	$C_{\mathcal{H}_{11}} = \{C_{17}, C_{18}\}$

**Table 5.7:** Definition of constraints and events.

$\mathcal{H}$	<i>PCTM</i>		<i>PCMC</i>		
	$[P(\mathcal{H})]$	T (ms)	$\widehat{P}(\mathcal{H})$	T (ms)	$\mathcal{E}$ (%)
$\mathcal{H}_9$	[0.9105,0.9115]	180	0.9110	1740	0
$\mathcal{H}_{10}$	[0.4978,0.4988]	130	0.4982	570	0
$\mathcal{H}_{11}$	[0.2486,0.2496]	50	0.2490	80	0

**Table 5.8:** Probability enclosures for events  $\mathcal{H}_i$  with PDF  $f_1$ , obtained with both versions of algorithm 6 and respective timings.

$\mathcal{H}$	<i>PCTM</i>		<i>PCMC</i>		
	$[P(\mathcal{H})]$	T (ms)	$\widehat{P}(\mathcal{H})$	T (ms)	$\mathcal{E}$ (%)
$\mathcal{H}_9$	[0.4472,0.4482]	120	0.4478	560	0
$\mathcal{H}_{10}$	[0.4995,0.5004]	310	0.5000	1740	0
$\mathcal{H}_{11}$	[0.2103,0.2113]	100	0.2108	260	0

**Table 5.9:** Probability enclosures for events  $\mathcal{H}_i$  with PDF  $f_2$ , obtained with both versions of algorithm 6 and respective timings.

The expected value of  $\mathbf{Z} = \langle X, Y \rangle$  and its covariance matrix, when the joint PDF of  $\mathbf{Z}$  is  $f_2$ , computed by the PCTM version of algorithm 10, with parameters  $\varepsilon = 10^{-15}$ ,  $\delta = 0.001$  and a Taylor model order of 2, are:

$$E[\mathbf{Z}] \in \langle [-0.0013, 0.0013], [-0.2471, -0.2447] \rangle$$

$$\Sigma[\mathbf{Z}] \in \begin{bmatrix} [3.2846, 3.2951] & [-0.0030, 0.0030] \\ [-0.0030, 0.0030] & [2.8643, 2.8741] \end{bmatrix}$$

and for the PCMC version, with parameters  $\varepsilon = 10^{-15}$ ,  $\delta = 0.005$  and  $N = 100$  random sample points, are:

$$\widehat{E}[\mathbf{Z}] = \langle -0.00004, -0.2460 \rangle$$

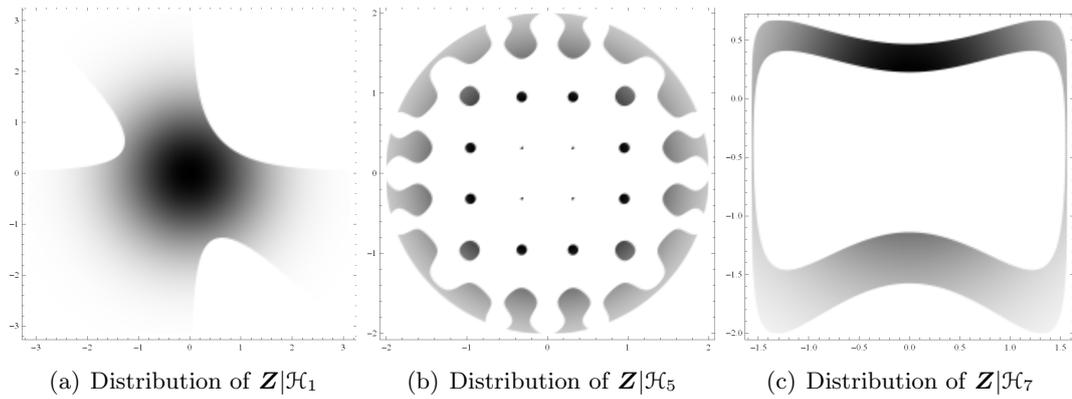
$$\widehat{\Sigma}[\mathbf{Z}] = \begin{bmatrix} 3.2899 & 0.00005 \\ 0.00005 & 2.8692 \end{bmatrix}$$

### 5.6.3 Probability Distributions

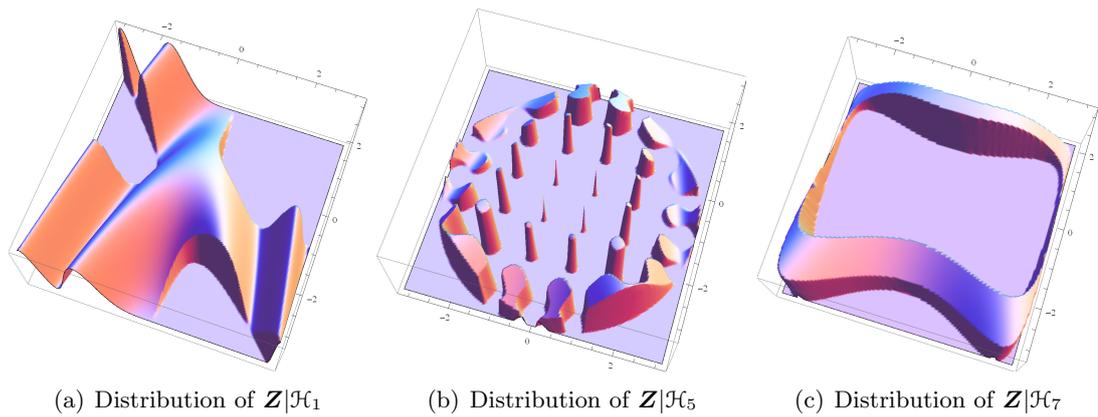
The figures for the probability distributions presented in this subsection are based on the midpoints of the interval enclosures for the grid boxes probabilities returned by algorithms 12 and 13 in array  $M$ . Due to the similarity of the results obtained with both versions of the algorithms, only those obtained with the PCTM version are presented here.

Figures 5.13 and 5.14 show the distributions of  $\mathbf{Z} = \langle X, Y \rangle$  conditioned by  $\mathcal{H}_1$ ,  $\mathcal{H}_5$  and  $\mathcal{H}_7$ , computed by algorithm 12 for, respectively,  $\langle CP, f_1 \rangle$  and  $\langle CP, f_2 \rangle$  with  $L = \langle 200, 200 \rangle$  and  $T = 30$  ms (for step 1) and  $\varepsilon = 10^{-15}$  and  $\delta = 0.001$  (for step 3) and a Taylor order of 2. In figure 5.13, darker colors represent more likely regions. Figures 5.15 and 5.16 show marginal distributions of  $X$  and  $Y$  for, respectively,  $\langle CP, f_1 \rangle$  and  $\langle CP, f_2 \rangle$  with the same parameters. The time spent to compute the joint box covers to obtain the presented distributions was less than 1 minute for all events and both PDFs.

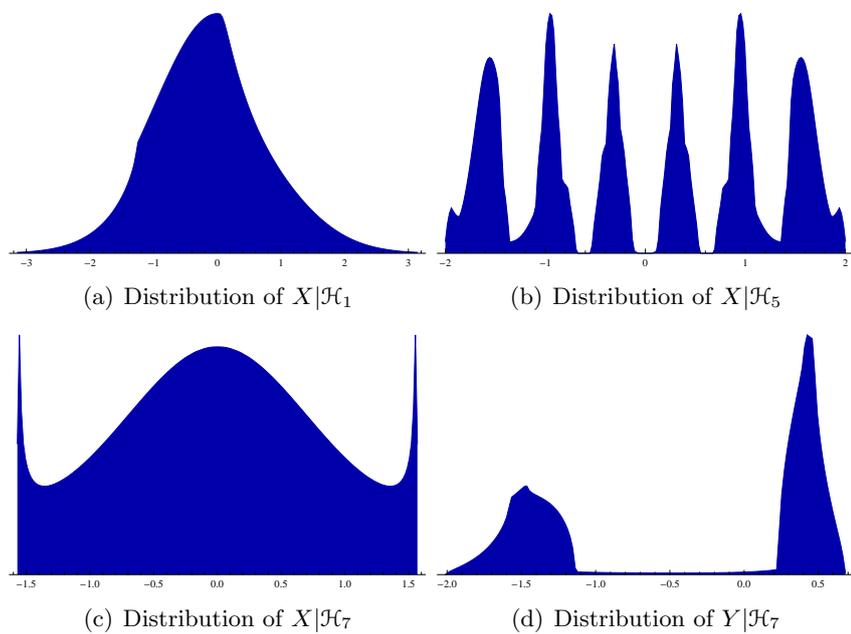
Due to the symmetry characteristics of events  $\mathcal{H}_1$  and  $\mathcal{H}_5$  and PDF  $f_1$ , the distributions for  $Y|\mathcal{H}_1$  and for  $Y|\mathcal{H}_5$  (not shown in figure 5.15) are equivalent, respectively, to those for  $X|\mathcal{H}_1$  and for  $X|\mathcal{H}_5$ .



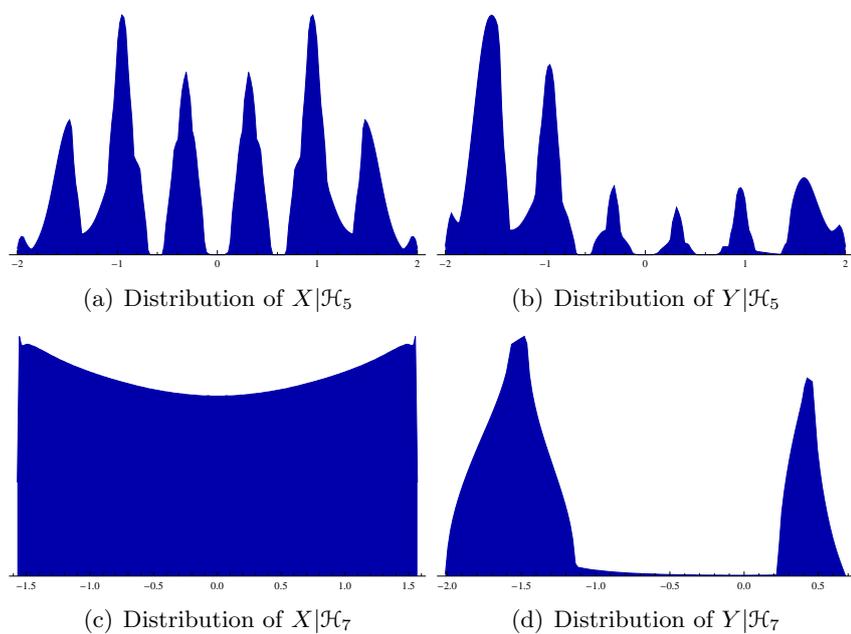
**Figure 5.13:** Probability distributions of  $\mathbf{Z} = \langle X, Y \rangle$  conditioned by events  $\mathcal{H}_i$  obtained with algorithm 12 with PDF  $f_1$ .



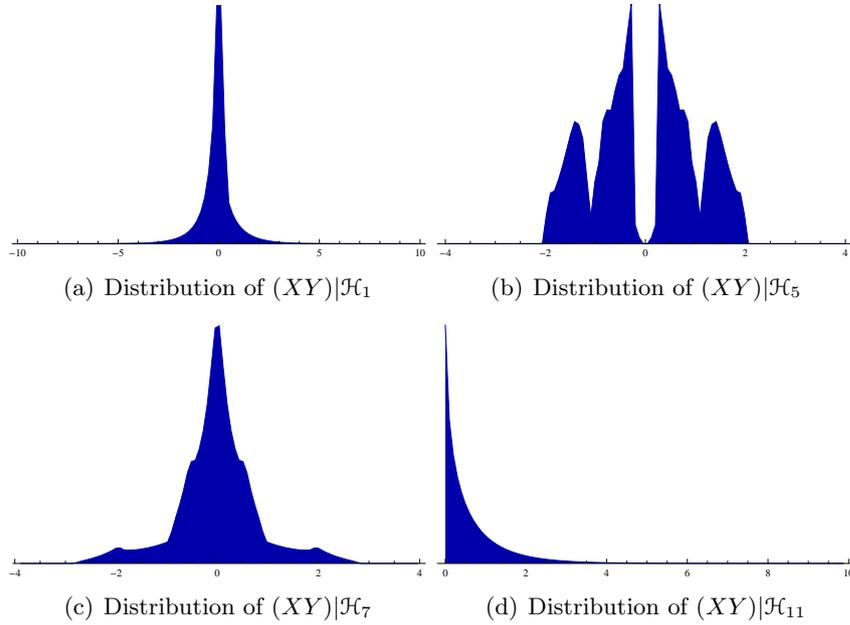
**Figure 5.14:** Probability distributions of  $\mathbf{Z} = \langle X, Y \rangle$  conditioned by events  $\mathcal{H}_i$  obtained with algorithm 12 with PDF  $f_2$ .



**Figure 5.15:** Marginal probability distributions of  $X$  and  $Y$  conditioned by events  $\mathcal{H}_i$  obtained with algorithm 12 with PDF  $f_1$ .



**Figure 5.16:** Marginal probability distributions of  $X$  and  $Y$  conditioned by events  $\mathcal{H}_i$  obtained with algorithm 12 with PDF  $f_2$ .



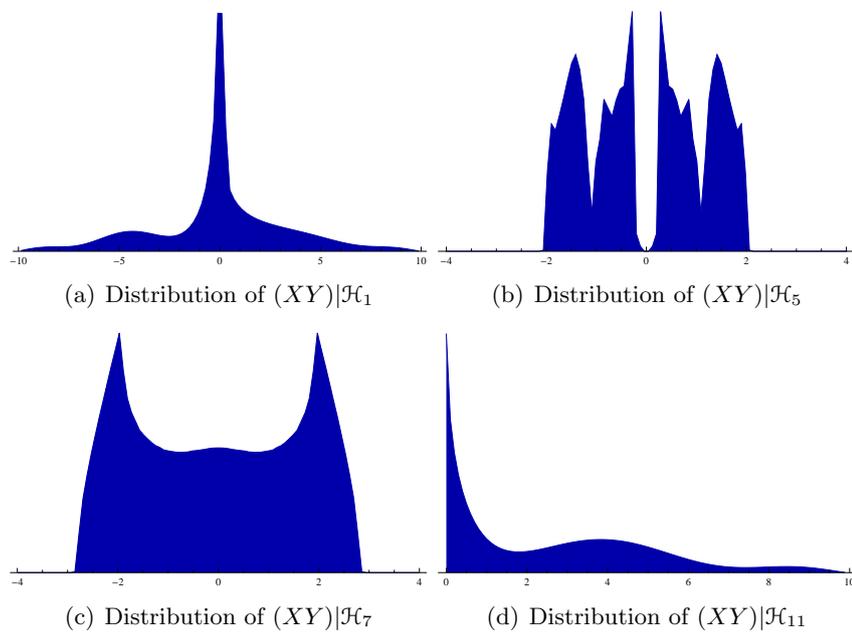
**Figure 5.17:** Probability distributions of  $XY$  conditioned by PC events obtained with algorithm 12 with PDF  $f_1$ .

Figures 5.17 and 5.18 show the distributions of  $\langle XY \rangle$  conditioned by events  $\mathcal{H}_1$ ,  $\mathcal{H}_5$ ,  $\mathcal{H}_7$  and  $\mathcal{H}_{11}$ , computed by algorithm 13 for, respectively,  $\langle CP, f_1 \rangle$  and  $\langle CP, f_2 \rangle$  with  $L = \langle 100 \rangle$  and  $T = 30$  ms (for step 1) and  $\varepsilon = 10^{-15}$  and  $\delta = 0.01$  (for step 3) and a Taylor order of 2.

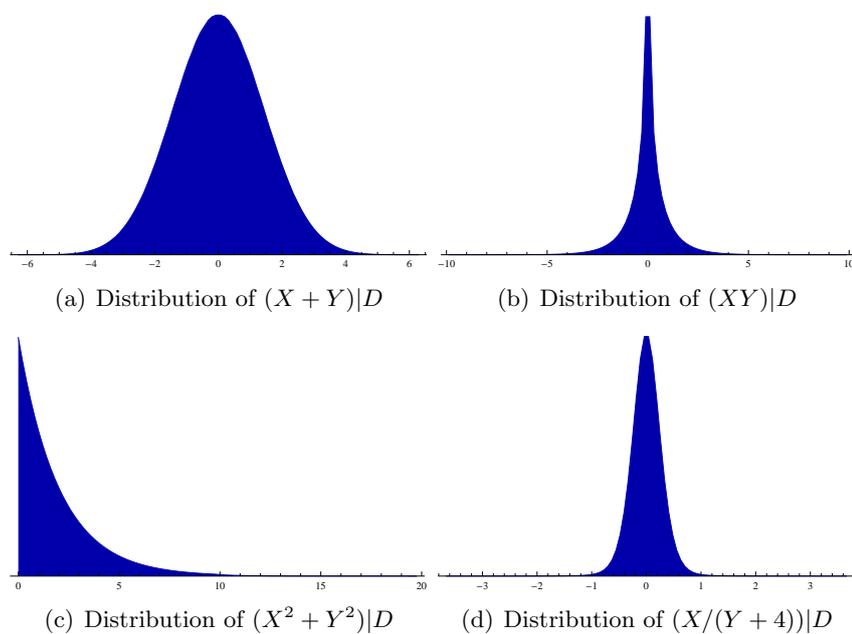
Figures 5.19 and 5.20 show the distributions of  $\langle X+Y \rangle$ ,  $\langle XY \rangle$ ,  $\langle X^2+Y^2 \rangle$  and  $X/(Y+4)$  conditioned by event  $D$ , computed by algorithm 13 for, respectively,  $\langle CP, f_1 \rangle$  and  $\langle CP, f_2 \rangle$  with the same parametrization as above, except for  $\delta = 0.05$ . As expected, the sum of two normally distributed random variables has a normal shaped distribution (figure 5.19 (a)).

The impact of the conditioning events  $\mathcal{H}_1$ ,  $\mathcal{H}_5$  and  $\mathcal{H}_7$  on the distribution of  $XY$  (figures 5.17 and 5.18) when compared with the unconditional case (i.e., conditioned only by  $D$ , in figures 5.19 (b) and 5.20 (b)) is more noticeable for events  $\mathcal{H}_5$  and  $\mathcal{H}_7$ , since the region missed by event  $\mathcal{H}_1$  has a small contribution to the overall probability.

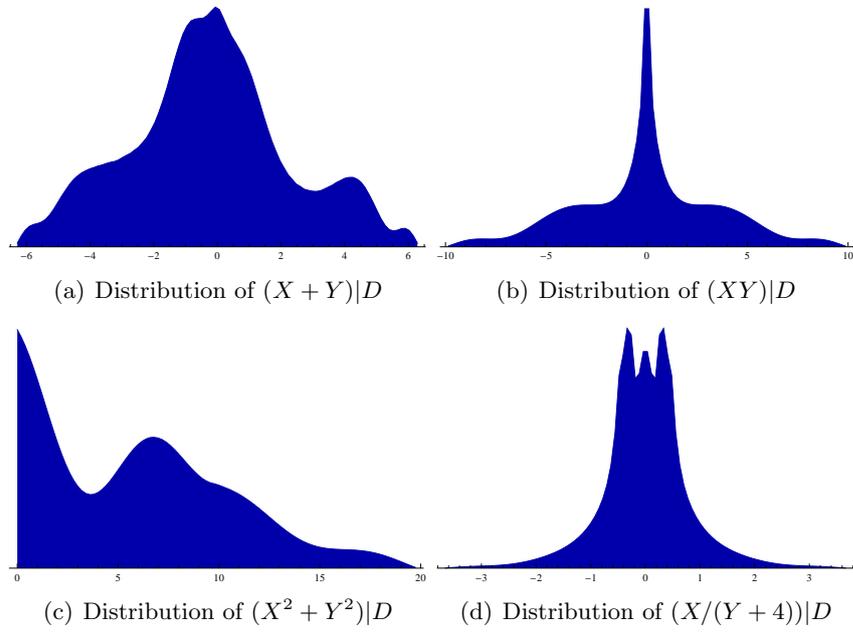
Finally, figure 5.21 presents the distributions of  $\langle X+Y, X \rangle$  conditioned by event  $D$ , computed by the algorithm that combines algorithms 12 and 13, for  $\langle CP, f_2 \rangle$  with the same parametrization as above.



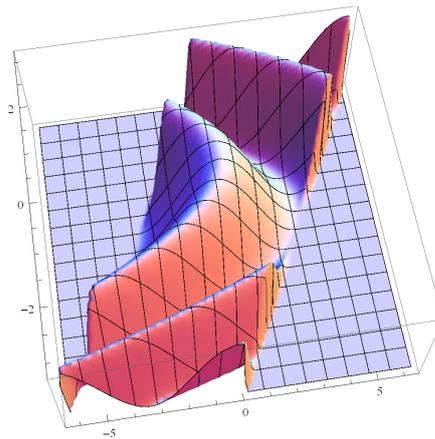
**Figure 5.18:** Probability distributions of  $XY$  conditioned by PC events obtained with algorithm 12 with PDF  $f_2$ .



**Figure 5.19:** Probability distributions conditioned by  $D$  obtained with algorithm 12 with PDF  $f_1$ .



**Figure 5.20:** Probability distributions conditioned by  $D$  obtained with algorithm 12 with PDF  $f_2$ .



**Figure 5.21:** Probability distribution of  $\langle X+Y, X \rangle | D$  with PDF  $f_2$ .

		PCTM		PCMC	
$\mathcal{H}$	$x_i$	$[E[x_i \mathcal{H}]]$	$[Var[x_i \mathcal{H}]]$	$\widehat{E}[x_i \mathcal{H}]$	$\widehat{Var}[x_i \mathcal{H}]$
$\mathcal{H}_1$	$x$	[-0.1305,-0.1292]	[0.7818,0.7849]	-0.1299	0.7836
	$y$	[-0.1305,-0.1292]	[0.7818,0.7849]	-0.1297	0.7836
$\mathcal{H}_5$	$x$	[-0.0046, 0.0046]	[1.3815,1.4065]	-0.0005	1.3912
	$y$	[-0.0046, 0.0046]	[1.3815,1.4065]	-0.0013	1.3934
$\mathcal{H}_7$	$x$	[-0.0035, 0.0035]	[0.6271,0.6404]	0.0098	0.6396
	$y$	[-0.3526,-0.3443]	[0.8380,0.8578]	-0.3535	0.8525

**Table 5.10:** Expected values and variances conditioned by events  $\mathcal{H}_i$  obtained by PCTM and PCMC versions, with PDF  $f_1$ .

$\mathcal{H}$	$Z$	$[E[Z \mathcal{H}]]$	$[Var[v \mathcal{H}]]$	$[Cov(X, Y)]$
$\mathcal{H}_1$	$X$	[-0.3856, -0.3821]	[2.7446, 2.7581]	[-0.2166, -0.2078]
	$Y$	[-0.6153, -0.6117]	[2.0964, 2.1092]	
$\mathcal{H}_5$	$X$	[-0.0084, 0.0084]	[1.0592, 1.0989]	[-0.0138, 0.01378]
	$Y$	[-0.6793, -0.6470]	[1.3513, 1.4401]	
$\mathcal{H}_7$	$X$	[-0.0058, 0.0058]	[0.8626, 0.8869]	[-0.0121, 0.0121]
	$Y$	[-1.0121, -0.9885]	[0.7694, 0.8308]	

**Table 5.11:** Enclosures for expected values and covariances conditioned by events  $\mathcal{H}_i$  obtained with PCTM version, with PDF  $f_2$ .

#### 5.6.4 Conditional Probabilistic Enclosures

The joint box covers produced in the experiments of the previous section (to compute conditional probability distributions of  $\mathbf{Z} = \langle X, Y \rangle$  given events  $\mathcal{H}_1$ ,  $\mathcal{H}_5$  and  $\mathcal{H}_7$  with the PCTM version of algorithm 12) were used to compute conditional expected values and variances of  $X$  and  $Y$ . Those values, as well as the ones obtained with the PCMC version of the algorithm, are shown in table 5.10 for PDF  $f_1$  and in tables 5.11 and 5.12 for PDF  $f_2$ . All approximations computed by the PCMC version have an very small error percentage (almost always 0%) when compared with the enclosures computed by the PCTM version. Time spent to compute the moments, after the computation of the joint box covers, was less than 15 seconds in the PCTM version, while in the PCMC version was less than 5 seconds, for all events and both PDFs.

$\mathcal{H}$	$Z$	$\widehat{E}[Z \mathcal{H}]$	$\widehat{V}_{ar}[Z \mathcal{H}]$	$[\widehat{Cov}(X, Y)]$
$\mathcal{H}_1$	$X$	-0.3831	2.7513	-0.2113
	$Y$	-0.6138	2.1017	
$\mathcal{H}_5$	$X$	-0.0040	1.0782	0.0021
	$Y$	-0.6619	1.3951	
$\mathcal{H}_7$	$X$	-0.0036	0.8770	0.0048
	$Y$	-0.9963	0.8018	

**Table 5.12:** Approximations for expected values and covariances conditioned by events  $\mathcal{H}_i$  obtained with PCMC version, with PDF  $f_2$ .

## 5.7 Summary

The main focus of this chapter was on probabilistic features of random vectors within a probabilistic continuous constraint space. It presented methods to compute safe and approximate enclosures for probabilities (conditional or not) of random vectors when restricted to a range of values, as well as for unconditional and conditional expected values and covariance matrices of random vectors. It also presented methods to compute probability distributions of a subset of the identity random vector and of random vectors defined as functions of the former. Experimental results illustrated the capabilities of the proposed algorithms.

The third part of the dissertation presents the application of the proposed framework to decision problems, showing how they can be cast as probabilistic continuous constraint spaces and using methods proposed in part II to solve them. Comparisons are made with the classical techniques to solve that kind of problems. In particular, the next chapter focuses on nonlinear inverse problems.



## Part III

# Application to Decision Problems



## CHAPTER 6

# Nonlinear Inverse Problems

Many problems of practical interest can be formulated as nonlinear inverse problems [112]. Such problems aim at finding the parameters of a model, given by systems of equations, from noisy data. These are typically ill-posed problems that may have no exact solutions, multiple solutions or unstable solutions.

Classical approaches for these problems are based on nonlinear regression methods [11] which search for the model parameter values that best-fit a given criterion. Best-fit approaches, often based on local search methods, provide single scenarios that may be inadequate to the characterization of the parameters.

In contrast, continuous constraint programming provides a framework to characterize the set of all scenarios consistent with the constraints of a problem given the uncertainty on its parameters. This is achieved through constraint reasoning, where initial intervals, representing the uncertainty on parameter values, are safely narrowed by reliable interval methods. Nevertheless, the application of classical constraint approaches to nonlinear inverse problems [53, 68] suffer from a major pitfall of considering the same likelihood for all values in the intervals.

In this chapter we show how the probabilistic constraint (PC) framework can be used as an effective tool for dealing with nonlinear inverse problems and we illustrate its main features in three application problems. The first two problems, a model for the propagation of seismic waves and a model for the population growth, are drawn from the literature on inverse problems and illustrate how the PC framework can deal with nonlinear inverse problems, in general. The last problem is more complex, addressing

ocean color inversion, and is intended to show that the PC framework can be used in real world applications.

## 6.1 Inverse Problems

Inverse problems aim to estimate parameters from observed data based on a model of the system behavior. The model variables are divided into model parameters,  $\mathbf{m} = (m_1, \dots, m_n)$ , whose values completely characterize the system and observable parameters,  $\mathbf{o} = (o_1, \dots, o_k)$ , that can be measured. The model is typically a forward mapping  $\mathbf{g}$  from the model parameters to the observable parameters and allows to predict the results of measurements based on the model parameters.

$$\mathbf{o} = \mathbf{g}(\mathbf{m}) \tag{6.1}$$

In the inverse theory literature several authors use the term inverse problem to refer exclusively to continuous inverse problems, i.e. inverse problems where the observations and/or the parameters are described by functions. Nevertheless, other authors (see, for example, [88, 112]) classify inverse problems depending on the way the parameters and observations are described. If both are discrete, i.e., described by a vector of values, they are discrete inverse problems or parameter estimation problems. These are the problems we consider in the present work and we simply call them inverse problems.

Uncertainty arises from measurement errors on the observed data or approximations in the model specification. When the model equations  $\mathbf{g}$  are nonlinear, the problem is a nonlinear inverse problem. Nonlinearity and uncertainty play a major role in modeling the behavior of most real systems.

Nonlinear inverse problems are typically ill-posed problems: they may have no exact solutions (no combination of parameter values is capable of exactly predicting all the observed data), solutions are not necessarily unique (different combinations of parameter values may lead to the same observable values) and the stability of solutions is not guaranteed (small changes in the observed data may result in arbitrarily large changes in the model parameters).

## 6.2 Classical Techniques

In [88] Menke states that *there are many different points of view regarding what constitutes a solution to an inverse problem*. He also presents *some possible forms an answer to an inverse problem might take* such as *estimates, bounding values and probability distributions* of the model parameters. In this section we cover methods for these three approaches to inverse problems: nonlinear regression methods provide an estimate of the model parameters; bounded error estimation gives guaranteed bounding values for the model parameters; and other stochastic approaches provide *a posteriori* probability distributions of the model parameters.

Classical approaches for addressing nonlinear inverse problems are based on nonlinear regression methods [11] which search for the model parameter values that best-fit a given criterion. A regression model specifies a dependent variable  $y$  by a function of one or more independent variables  $x^1$  and model parameters  $\mathbf{m}$ :

$$y \approx f(x, \mathbf{m}) \quad (6.2)$$

If a set of  $k$  observations of the system is known  $\{\langle x_i, y_i \rangle\}$  and a random error  $\varepsilon_i$  is assumed to represent the measurement error around the observed quantity  $y_i$ , the regression model becomes:

$$y_i = f(x_i, \mathbf{m}) + \varepsilon_i, \quad 1 \leq i \leq k \quad (6.3)$$

Equations (6.1) and (6.3) are similar if we consider  $o_i = y_i$  and  $g_i(\mathbf{m}) = f(x_i, \mathbf{m})$ , except that the error term  $\varepsilon_i$  appears explicit in (6.3). Based on several assumptions on the distribution of the errors  $\varepsilon_i$ , nonlinear regression methods search for the model parameter values that minimize a suitable criterion. Moreover, with additional assumptions on the regression model, adequate analytic techniques may be used to characterize the uncertainty around the obtained parameter values.

---

<sup>1</sup>To simplify the notation we assume a single independent variable  $x$ . In the case of multiple independent variables,  $x$  should be replaced by a vector of variables  $\mathbf{x}$ .

For instance, the least squares criterion minimizes a quadratic norm of the difference between the vector of observed data and the vector of model predictions:

$$\sum_{i=1}^k (y_i - f(x_i, \mathbf{m}))^2 = \sum_{i=1}^k \varepsilon_i^2, \quad 1 \leq i \leq k \quad (6.4)$$

If errors  $\varepsilon_i$  are independent and normally distributed with zero mean and constant variance, then the least squares estimator is the maximum likelihood estimator (estimates values of the model parameters that produce a distribution that gives the observed data the greatest probability). Additionally, if  $f$  is linear with respect to the model parameters, then the values of the parameters given the observed data are necessarily normally distributed and confidence regions can be analytically computed.

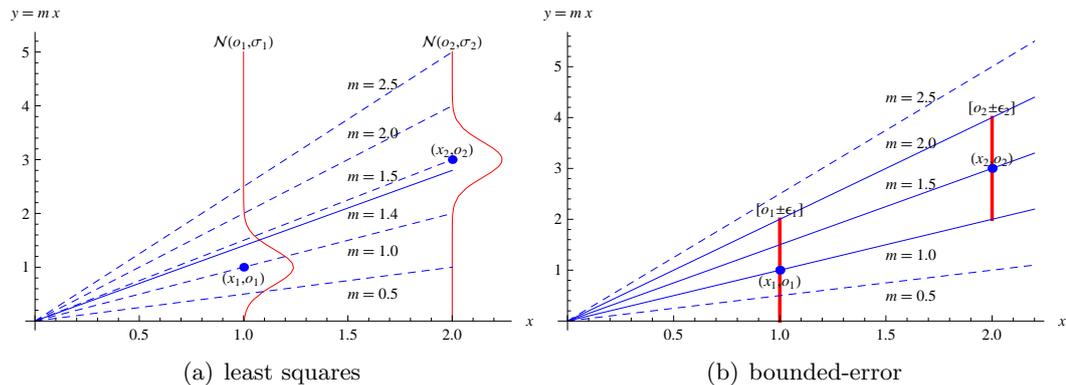
An alternative constraint approach is known as bounded-error estimation or set membership estimation [53, 68]. The idea is to replace the search for a single best-fit solution with the characterization of the set of all solutions consistent with acceptable measurement errors around the observations. Bounded-error estimation assumes reliable bounds for each measurement error  $\varepsilon_i$ , namely  $a_i \leq \varepsilon_i \leq b_i$ , and applies constraint solving techniques to compute the feasible space of:

$$y_i - b_i \leq f(x_i, \mathbf{m}) \leq y_i - a_i, \quad 1 \leq i \leq k \quad (6.5)$$

A safe enclosure of the feasible space provides insight on the remaining uncertainty about the model parameter values. However this approach has the major pitfall of considering the same likelihood for all consistent solutions.

Consider the example illustrated in figure 6.1 (a) of a simple linear regression model  $y_i = mx + \varepsilon_i$  with a single parameter  $m$  and 2 observations  $\langle x_1, y_1 \rangle = \langle 1, 1 \rangle$  and  $\langle x_2, y_2 \rangle = \langle 2, 3 \rangle$ . The measurement errors  $\varepsilon_1$  and  $\varepsilon_2$  are assumed to be independent and normally distributed with zero mean and standard deviation  $\sigma = 1/3$ . From the different possible values for parameter  $m$  the least squares method computes  $m^* = 1.4$  that maximizes the likelihood of the observed data (solid line). For a given value of parameter  $m$  the likelihood of each observed data is the probability of making a measurement error with a magnitude equal to the difference between the observed and

the predicted values (as illustrated in the figure for  $m^*$ ). Since measurement errors are assumed independent the overall likelihood of the observed data is the product of such probabilities.

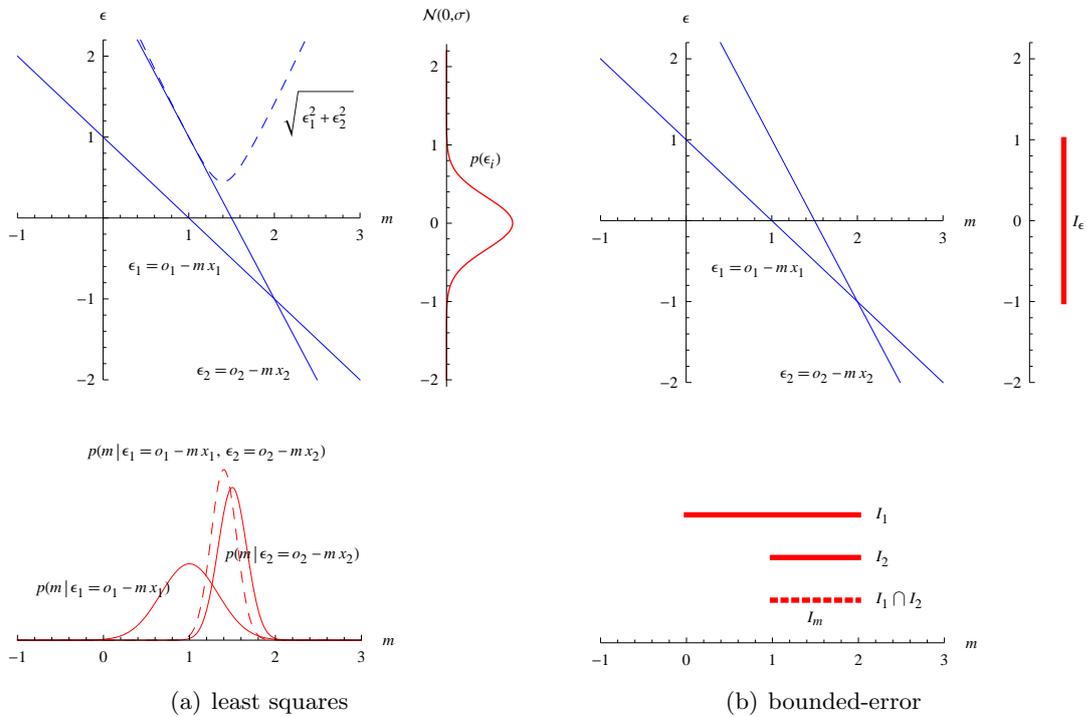


**Figure 6.1:** Least squares approaches a) search for a single best-fit solution that maximize the likelihood of the observations. Bounded-error approaches b) compute a set of solutions consistent with the observations.

Figure 6.1 (b) illustrates the bounded-error approach for the same problem. Assuming acceptable measurement errors between  $\pm 1$  around the observed values ( $\pm 3\sigma$  in the previous approach) only some  $m$ -values (corresponding to all the lines that cross both vertical segments) are consistent with the model and the observations. The result of the bounded-error estimation is that  $m \in [1, 2]$  without providing any specification of the values distribution.

Figure 6.2 (a) illustrates the probability distribution of the parameter values given the observed data and the regression model of the previous example. Each straight line concerns a single observation  $\langle x_i, y_i \rangle$  and represents the measurement error  $\varepsilon_i$  as a function of the parameter value  $m$  (these are straight lines because the model is linear with respect to  $m$ ). The dashed curve in the upper graphic represents the least squares criterion (square root) as a function of the parameter value  $m$ . Clearly, such function has a single minimum, which in this case is at  $m = 1.4$ . The right hand side of the upper graphic shows the measurement error distribution (equal for each measurement). The solid curves in the lower graphic represent the probability distribution of the parameter values given each observation. They are obtained by computing to each  $m$ -value the  $\varepsilon$ -value of the respective line and then assigning the probability of such error (informally: go straight vertical up to the respective line and

then straight right to get the probability value in the upper right graphic). After normalization, the obtained curves are necessarily normal distributions since the slopes of the straight lines determine the mean and the variance of the distribution but cannot reshape the error distributions into non normal distributions. The dashed curve in the lower graphic represents the probability distribution of the parameter values given all the observations and is obtained by pointwise multiplication of the solid curves (and posterior normalization). Clearly, it must be also a normal distribution and, as expected, its mean value is exactly the  $m$ -value that minimizes the least squares criterion.



**Figure 6.2:** In linear problems, least squares approaches a) compute the mean value of the distribution of the parameter values given the observed data. Bounded-error approaches b) compute an interval that includes all  $m$ -values consistent with the observations.

Figure 6.2 (b) is similar to figure 6.2 (a) but instead of considering probability distributions for the measurement errors only reliable bounds are assumed. In this case, the results are no longer normal probability distributions but single intervals representing all possible parameter values consistent with the observations.

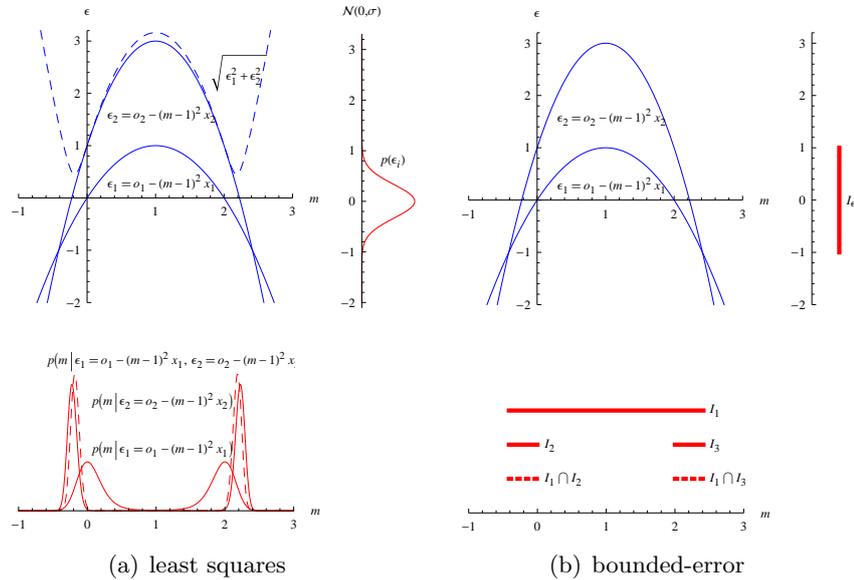
As seen in the previous figures, least squares approaches may be adequate for char-

acterizing the uncertainty on the model parameters given the observations. This is the case for linear and some weakly nonlinear problems, where efficient computational techniques exist to solve them as curve fitting problems.

However, in generic nonlinear problems, best-fit approaches may be inadequate for several reasons. Due to nonlinearity, the minimization function may have multiple local and global minima. Without an explicit formula for obtaining the best-fit values, minimization is usually performed through local search algorithms. If such algorithms converge to local minima, the solution obtained is no longer a maximum likelihood solution (there are solutions more likely than these provided by the method). Even when a maximum likelihood solution is found, the probability distribution of the model parameters given the observations may not be realistically approximated by a normal distribution and so the computation of any confidence regions based on such assumption will be inadequate.

Consider the previous example but now with the nonlinear regression model  $y_i = (m - 1)^2x + \varepsilon_i$ , as shown in figure 6.3. Note that the model is linear with respect to  $x$ , and so, each  $m$ -value determines a straight line similar to those in figure 6.1 (the difference is that a line with slope  $s$  is obtained now with  $m = 1 \pm \sqrt{s}$ ). In particular, if in the previous case the best-fit value was  $m^* = 1.4$  now it must be  $m^* = 1 \pm \sqrt{1.4}$ .

Figure 6.3 (a) is analogous to figure 6.2 (a) with the model replaced by the above nonlinear regression model. Now, each curve representing the measurement error  $\varepsilon_i$  as a function of the parameter value  $m$ , induces a bimodal probability distribution for the parameter values. Consequently, the probability distribution of the parameter values given the observations (dashed curve in the lower graphic) is far from being a normal distribution. From the dashed curve in the upper graphic it is clear that there are 2 global minima (at maximum likelihood values  $m^* = 1 \pm \sqrt{1.4}$ ). If a least squares method is applied to this problem it will eventually converge to one of these minimums and any inferences about confidence regions would be based on the assumption that such minimum is the mean value of a normal distribution for the parameter values which is completely wrong in this case. Note that this would not be the case if the domain of  $m$  is restricted to  $[2, 3]$  since in this region the curves in the upper graphic can be reasonably approximated by straight lines and so, the resulting distributions are nearly Gaussian.



**Figure 6.3:** In nonlinear problems, least squares approaches a) may provide wrong results assuming the computed value to be the mean of a parameter normal distribution given the observed data. Bounded-error approaches b) still provide reliable enclosures for all  $m$ -values consistent with the observations.

Figure 6.3b) illustrates that the bounded-error approach is robust with respect to non-linearity in the sense that consistent solutions are not lost and the resulting 2 enclosures provide a reliable characterization of the consistent parameter values.

Other stochastic alternatives to deal with nonlinear problems that are inadequate to best-fit approaches, associate an explicit probabilistic model to the problem [112]. In these approaches, prior information on the model parameters is represented by a probability distribution, which is transformed into an *a posteriori* probability distribution, by incorporating the forward model and the actual result of the observations (with their uncertainties). They typically rely on extensive random (Monte Carlo) sampling to characterize the *a posteriori* distributions of the parameter values (such as those in figure 6.3a)). However, as explained in section 6.6.4, pure Monte Carlo techniques, contrary to constraint approaches, cannot prune the sampling space based on model information, and this may be a significant drawback, specially in nonlinear problems.

## 6.3 Probabilistic Constraint Approach

The PC approach extends the reliable constraint framework, robust to nonlinearity, with a stochastic representation of the uncertainty on the parameters. Similarly to the stochastic approaches it associates an explicit probabilistic model to the problem, and similarly to error-bounded approaches it assumes reliable bounds for the measurement errors.

To use the PC framework it is thus necessary to specify a joint probability density function on the parameters taking into account the knowledge about the forward model and the observed data. For that purpose, we consider the joint PDF,  $p_{\mathbf{M},\mathbf{E}}(\mathbf{m},\boldsymbol{\varepsilon})$ , representing the initial knowledge about the parameter and measurement error distributions without including the forward model and the observations. By conditioning it to the event defined by the forward model instantiated with the observed data, we obtain the desired joint PDF, as presented next.

**Property 6.1 (Marginal Conditional PDF for the Parameters)** *Consider a joint PDF,  $p_{\mathbf{M},\mathbf{E}}(\mathbf{m},\boldsymbol{\varepsilon})$  for the multivariate random variables  $\mathbf{M}$  and  $\mathbf{E}$  that represent, respectively, the parameters and the measurement errors. The marginal conditional PDF for the multivariate random variable  $\mathbf{M}$  given the event  $\mathcal{H} = \{\langle \mathbf{v}_m, v_{\varepsilon_1}, \dots, v_{\varepsilon_k} \rangle \in \mathbb{R}^{n+k} : \forall_{1 \leq i \leq k} v_{\varepsilon_i} = y_i - f(x_i, \mathbf{v}_m)\}$  is:*

$$p_{\mathbf{M}|\mathcal{H}}(\mathbf{m}) = \alpha p_{\mathbf{M},\mathbf{E}}(\mathbf{m}, y_1 - f(x_1, \mathbf{m}), \dots, y_k - f(x_k, \mathbf{m}))$$

where the  $x_i$  and  $y_i$  are constants and  $\alpha$  is a normalization constant:

$$\alpha = 1 / \int_{\mathbb{R}^n} p_{\mathbf{M},\mathbf{E}}(\mathbf{m}, y_1 - f(x_1, \mathbf{m}), \dots, y_k - f(x_k, \mathbf{m})) d\mathbf{m}$$

**Proof.** For simplicity we consider a single observation  $\langle x, y \rangle$  with measurement error  $\varepsilon$ , but the proof can be extended to multiple observations, by considering a multiple integral. The symbol  $\int_{-\infty}^{v_m}$  is a multiple integral (with  $n$  dimensions) where the  $i^{th}$  integral is between  $-\infty$  and the  $i^{th}$  element from the  $n$ -tuple  $\mathbf{m}$ . The function  $h(\cdot)$  denotes  $y - f(x, \cdot)$ .

Consider the CDF  $F_{M|\mathcal{H}}(\mathbf{m})$  such that  $p_{M|\mathcal{H}}(\mathbf{m}) = \frac{\partial^n F_{M|\mathcal{H}}}{\partial \mathbf{m}}(\mathbf{m})$ . Let us expand the CDF formula.

$$\begin{aligned} F_{M|\mathcal{H}}(\mathbf{m}) &= P(\mathbf{M} \leq \mathbf{m} | \mathcal{H}) = \frac{P(\{\mathbf{M} \leq \mathbf{m}\} \cap \mathcal{H})}{P(\mathcal{H})} \\ &= \lim_{\delta \rightarrow 0} \frac{\int_{-\infty^n}^{\mathbf{m}} \int_{h(\mathbf{v})-\delta}^{h(\mathbf{v})+\delta} p_{M,\mathbf{E}}(\mathbf{v}, \varepsilon) d\varepsilon d\mathbf{v}}{\int_{\mathbb{R}^n} \int_{h(\mathbf{v})-\delta}^{h(\mathbf{v})+\delta} p_{M,\mathbf{E}}(\mathbf{v}, \varepsilon) d\varepsilon d\mathbf{v}} \end{aligned}$$

By variable substitution  $\varepsilon = 2u\delta + h(\mathbf{v}) - \delta$  and, consequently  $d\varepsilon = 2\delta du$ , we obtain

$$\begin{aligned} &\lim_{\delta \rightarrow 0} \frac{\int_{-\infty^n}^{\mathbf{m}} \int_0^1 p_{M,\mathbf{E}}(\mathbf{v}, 2u\delta + h(\mathbf{v}) - \delta) 2\delta du d\mathbf{v}}{\int_{\mathbb{R}^n} \int_0^1 p_{M,\mathbf{E}}(\mathbf{v}, 2u\delta + h(\mathbf{v}) - \delta) 2\delta du d\mathbf{v}} \\ &= \lim_{\delta \rightarrow 0} \frac{2\delta \int_{-\infty^n}^{\mathbf{m}} \int_0^1 p_{M,\mathbf{E}}(\mathbf{v}, 2u\delta + h(\mathbf{v}) - \delta) du d\mathbf{v}}{2\delta \int_{\mathbb{R}^n} \int_0^1 p_{M,\mathbf{E}}(\mathbf{v}, 2u\delta + h(\mathbf{v}) - \delta) du d\mathbf{v}} \\ &= \frac{\int_{-\infty^n}^{\mathbf{m}} \int_0^1 p_{M,\mathbf{E}}(\mathbf{v}, h(\mathbf{v})) du d\mathbf{v}}{\int_{\mathbb{R}^n} \int_0^1 p_{M,\mathbf{E}}(\mathbf{v}, h(\mathbf{v})) du d\mathbf{v}} = \frac{\int_{-\infty^n}^{\mathbf{m}} p_{M,\mathbf{E}}(\mathbf{v}, h(\mathbf{v})) d\mathbf{v}}{\int_{\mathbb{R}^n} p_{M,\mathbf{E}}(\mathbf{v}, h(\mathbf{v})) d\mathbf{v}} \end{aligned}$$

Making  $\alpha = 1 / \int_{\mathbb{R}^n} p_{M,\mathbf{E}}(\mathbf{v}, h(\mathbf{v})) d\mathbf{v}$ , we get

$$F_{M|\mathcal{H}}(\mathbf{m}) = \alpha \int_{-\infty^n}^{\mathbf{m}} p_{M,\mathbf{E}}(\mathbf{v}, h(\mathbf{v})) d\mathbf{v}$$

By derivation we obtain  $p_{M|\mathcal{H}}(\mathbf{m}) = \frac{\partial^n F_{M|\mathcal{H}}}{\partial \mathbf{m}}(\mathbf{m}) = \alpha p_{M,\mathbf{E}}(\mathbf{m}, h(\mathbf{m}))$  and thus property 6.1 holds.  $\blacksquare$

We are now able to define how to model an inverse problem as a PC.

**Definition 6.1 (Inverse Problem as a Probabilistic Constraint Space)**

Consider an inverse problem with  $n$  model parameters  $m_1, \dots, m_n$  whose range is  $D \in \mathbb{I}\mathbb{R}^n$ ,  $k$  observations  $\langle x_1, y_1 \rangle, \dots, \langle x_k, y_k \rangle$  having measurement errors  $\varepsilon_1, \dots, \varepsilon_k$

with reliable bounds  $[a_i, b_i]$ ,  $1 \leq i \leq k$ , a forward model defined by the equation  $y = f(x, m_1, \dots, m_n)$  and a marginal conditional PDF  $p_{M|X}(\cdot)$  for the model parameters as defined in property 6.1. This inverse problem is modeled as a PC,  $\langle\langle X, D, C \rangle, p\rangle$ , such that:

$$D \in \mathbb{I}\mathbb{R}^n$$

$$X = \langle m_1, \dots, m_n \rangle$$

$$C = \{y_i - b_i \leq f(x_i, m_1, \dots, m_n) \leq y_i - a_i : 1 \leq i \leq k\}^1$$

$$p = p_{M|X}(\mathbf{m}) = \alpha p_{M,E}(\mathbf{m}, y_1 - f(x_1, \mathbf{m}), \dots, y_k - f(x_k, \mathbf{m}))$$

Together with the probabilistic model specified by the PDF, the PC approach considers the same set of constraints (6.5) enforced in the error-bounded approaches. The reliable bounds assumed for each measurement error may be tuned with their respective values distribution. For instance, if a Gaussian error distribution is assumed with zero mean and variance  $\sigma^2$ , then enforcing values to be within  $[-3\sigma, 3\sigma]$  captures about 99.7% of possibilities. In the limit, if all the error bounds are set to  $[-\infty, +\infty]$ , the method degenerates into a stochastic problem with no constraints to be enforced.<sup>2</sup>

When defining a joint PDF representing the parameter and measurement error distributions without including the forward model and the observations, there might be no reason to believe that some parameter values are more likely than others. Their joint initial distributions are typically uniform<sup>3</sup>. Assuming that an error in one measurement does not affect the error in any other measurement, without considering the forward model, they can be naturally assumed independent and so, their joint initial distributions are the product of their individual distributions. With such assumptions, a joint PDF for the parameters and measurement errors is:

$$p_{M,E}(\mathbf{m}, \varepsilon_1, \dots, \varepsilon_k) = \alpha_{\mathbf{m}} \prod_{i=1}^k p_i(\varepsilon_i)$$

<sup>1</sup>Notice that  $y_i - b_i \leq f(x_i, m_1, \dots, m_n) \leq y_i - a_i$  is an abbreviated way to represent the two constraints whose relations are defined by:  $\rho_{i_1} = \{d \in D : y_i - b_i - f(x_i, d) \leq 0\}$  and  $\rho_{i_2} = \{d \in D : f(x_i, d) - y_i + a_i \leq 0\}$ .

<sup>2</sup>The PC framework still requires initial bounds for the parameter values.

<sup>3</sup>This is not mandatory and other distributions can be considered.

where  $\alpha_{\mathbf{m}}$  is a constant (resulting from the uniform parameter distributions) and  $p_i$  is the PDF assumed for the measurement error  $\varepsilon_i$ . Notice that measurement error distributions may be non Gaussian but take any other form adequate to the instrumentation used in the respective observation. In the limit, the method degenerates into an error-bounded problem if all the measurement errors are considered uniform. Moreover, the independence assumption can be dropped as long as a joint PDF is explicitly specified as a function of the model parameters and the measurement errors.

Once established the PC  $\langle\langle X, D, C \rangle, p\rangle$  that represents the inverse problem, the PC approach relies on probabilistic constraint programming to compute any probabilistic information on the model parameters consistent with the experimental results. This is done by computing conditional probabilistic information, conditioned by the event  $\mathcal{H} = \mathcal{F}(\langle X, D, C \rangle)$ , the feasible space of the CCSP that models the inverse problem.

Given that conditioning is always used when computing probabilistic information associated with an inverse problem (e.g., marginal distributions for the parameters, expected values, covariance matrix) the PDF  $p$  can be used without the normalizing constant  $\alpha$  (see property 6.1), since it appears both on the numerator and denominator of the computations and so it will cancel. Consequently  $\alpha$  can be ignored without changing the computed values.

Conditional distributions of model parameters are computed using algorithm 12 (section 5.4, page 134) where  $\mathcal{H}$  is the conditioning event. The arguments for this algorithm are: the PC that models the inverse problem; the constraints associated with the conditioning event  $C_{\mathcal{H}} = C$ ; and the indices of the model parameters for which the conditional distribution will be computed,  $\mathbf{Z}_{idx}$ . The remaining arguments are defined case by case:  $\delta$ , which imposes the stopping criterion by defining the accuracy for the probability enclosure,  $\varepsilon$  used in the *eligible* $_{\varepsilon}$  predicate, and  $L$  (or  $\alpha$ ) that defines the number of partitions for the grid (or the grid spacings).

The PC associated with a given inverse problem can be easily extended to make predictions on the outcomes of new measurements. For this purpose a random variable that represents the new unknown observable parameter is defined as a function of the model parameters, using the forward model, i.e.,  $Y_i = g_i(\mathbf{m}) = f(x_i, \mathbf{m})$ . Then algorithm 13 (section 5.4, page 138) is used to compute the conditional distribution of each  $Y_i$  given

the event  $\mathcal{H}$ . The arguments of the algorithm are similar to those previously described (with  $G = \{[f](x_i, \mathbf{m})\}$ ).

The graphics for the probability distributions presented in the next sections are based on the midpoints of the interval enclosures for the grid boxes probabilities returned by algorithms 12 or 13 in array  $M$ .

In the case of the nonlinear example (figure 6.3) where the observations are  $\langle x_1, y_1 \rangle = \langle 1, 1 \rangle$  and  $\langle x_2, y_2 \rangle = \langle 2, 3 \rangle$ , the probabilistic model would be:

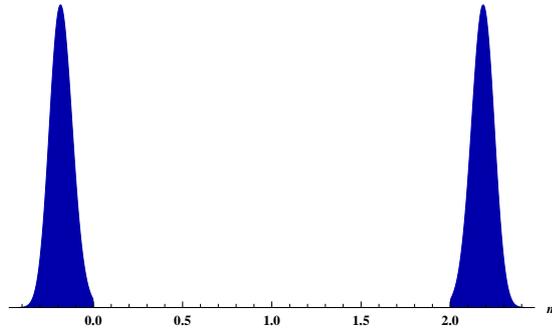
$$p(m) = \alpha \times \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{1 - 1 \times (m-1)^2}{\sigma} \right)^2} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{3 - 2 \times (m-1)^2}{\sigma} \right)^2}$$

and the constraints:

$$1 - 3\sigma \leq 1 \times (m-1)^2 \leq 1 + 3\sigma$$

$$3 - 3\sigma \leq 2 \times (m-1)^2 \leq 3 + 3\sigma$$

with  $\sigma = 1/3$ . Figure 6.4 illustrates the results obtained by the PC framework for the probability distribution of the parameter  $m$  conditioned by the above constraints, similar to the *a posteriori* distribution shown in figure 6.3 (a) (dashed line). Moreover, the PC framework is able to guarantee that all possible  $m$  values must be within the set  $[-0.4143, 0] \cup [2, 2.4143]$ , which agree with the results from the bounded error approach shown in figure 6.3 (b) (dashed line). Additionally, the enclosures obtained for the conditional expected value and variance are respectively  $[0.9999, 1.0001]$  and  $[1.3938, 1.3939]$ , which in face of the information provided in figure 6.4 are clearly not representative of the parameter distribution.



**Figure 6.4:** Probability distribution of the parameter  $m$  obtained by the PC framework.

## 6.4 Seismic Event Model

Consider the example of a nonlinear inverse problem extracted from [112]. The goal is to estimate the epicentral coordinates of a seismic event. The seismic waves produced have been recorded at a network of six seismic stations at different arrival times. Table 6.1 presents their coordinates and the observed arrival times.

$(x_i \text{ km}, y_i \text{ km})$	(3, 15)	(3, 16)	(4, 15)	(4, 16)	(5, 15)	(5, 16)
$t_i \text{ s}$	3.12	3.26	2.98	3.12	2.84	2.98

**Table 6.1:** Arrival times (in seconds) of the seismic waves observed at six seismic stations.

Clearly, the model parameters are the epicentral coordinates  $(m_1, m_2)$  of the seismic event, and the observable parameters are the six arrival times  $t_i$  which are related by a forward model with six equations (one for each seismic station  $i$ ):

$$t_i = f_i(x_i, y_i, m_1, m_2) = \frac{1}{v} \sqrt{(x_i - m_1)^2 + (y_i - m_2)^2} \quad (6.6)$$

It is assumed that: seismic waves travel at a constant velocity of  $v = 5 \text{ km/s}$ ; measurement errors on the arrival times are independent and are normally distributed  $\mathcal{N}(0, \sigma = 0.1)$  with reliable bounds within  $[-3\sigma, 3\sigma]$ ; the two parameters have initial ranges  $I_1$  and  $I_2$  and are uniformly distributed  $\mathcal{U}(I_j, \bar{I}_j)$ , with  $1 \leq j \leq 2$ .

The PDF  $p$  for the measurement errors is:

$$p_i(\varepsilon_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\varepsilon_i}{\sigma}\right)^2}$$

The joint PDF for the model parameters and measurement errors is:

$$p_{\mathbf{M}, \mathbf{E}}(m_0, m_1, \varepsilon_1, \dots, \varepsilon_6) = \alpha_{\mathbf{m}} \prod_{i=1}^6 p_i(\varepsilon_i) \quad (6.7)$$

where  $\alpha_{\mathbf{m}}$  is a constant resulting from the uniform parameter distributions.

This inverse problem is modeled as a PC  $\langle\langle X, D, C \rangle, p\rangle$  where:

$$\begin{aligned} X &= \langle m_1, m_2 \rangle \\ D &= I_1 \times I_2 \\ C &= \left\{ t_i - 3\sigma \leq \frac{1}{v} \sqrt{(x_i - m_1)^2 + (y_i - m_2)^2} \leq t_i + 3\sigma : 1 \leq i \leq 6 \right\} \\ p(m_1, m_2) &= \alpha \prod_{i=1}^6 p_i \left( t_i - \frac{1}{v} \sqrt{(x_i - m_1)^2 + (y_i - m_2)^2} \right) \quad \text{from property 6.1 and (6.7)} \end{aligned}$$

Figure 6.5 shows the conditional distribution of the model parameters, computed by algorithm 12, with  $D = [-100, 100] \times [-100, 100]$ ,  $\delta = 0.01$ , grid spacings  $\alpha = \langle 0.07, 0.07 \rangle$ ,  $Z_{idx} = \langle 1, 2 \rangle$  (i.e.,  $\mathbf{Z} = \langle m_1, m_2 \rangle$ ),  $\varepsilon = 10^{-15}$  and uses the validated quadrature method with a Taylor model order of 2.

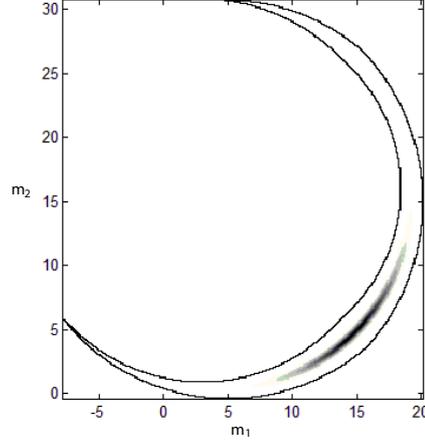
Besides identifying which combinations of  $m_1$  and  $m_2$  values are consistent, figure 6.5 illustrates its joint probability distribution, allowing to identify regions of maximum likelihood (darker colors represent more likely regions). An external contour was added to illustrate the safe enclosure of the feasible space obtained with classical constraint reasoning. Clearly the most likely region is concentrated in a much smaller area.

Figure 6.6 presents conditional probability distributions for each of the model parameters,  $m_1$  and  $m_2$ , computed by algorithm 12, with a parametrization similar to the one previously used, with (a)  $\mathbf{Z} = \langle m_1 \rangle$  and (b)  $\mathbf{Z} = \langle m_2 \rangle$ .

To compute the maximum likelihood point as in classical best-fit approaches, the PC framework can be embedded within an optimization algorithm that searches the maximum likelihood feasible point with guarantees of global optimality. For this example, it can be easily proved that the maximum likelihood point is in  $[14.70, 14.77] \times [4.65, 4.72]$ .

## 6.5 Population Growth Model

Another example of an inverse problem with a simple nonlinear forward model is a population growth model. Consider the data summarized in Table 6.2 based on the USA census over the years 1790 (normalized to 0) to 1910 with a 10 year period.



**Figure 6.5:** Epicentral coordinates of the seismic event. Joint distribution.

$t_i$	0	10	20	30	40	50	60	70	80	90	100	110	120
$y_i$	3.9	5.3	7.2	9.6	12.9	17.1	23.2	31.4	39.8	50.2	62.9	76.0	92.0

**Table 6.2:** US Population (in millions) over the years 1790 (0) to 1910 (120).

Assuming that an exponential growth is an acceptable model for the population growth, the forward model is defined by the set of equations (one for each observation  $i$ ):

$$y_i = f_i(t_i, m_1, m_2) = m_1 e^{m_2 t_i}$$

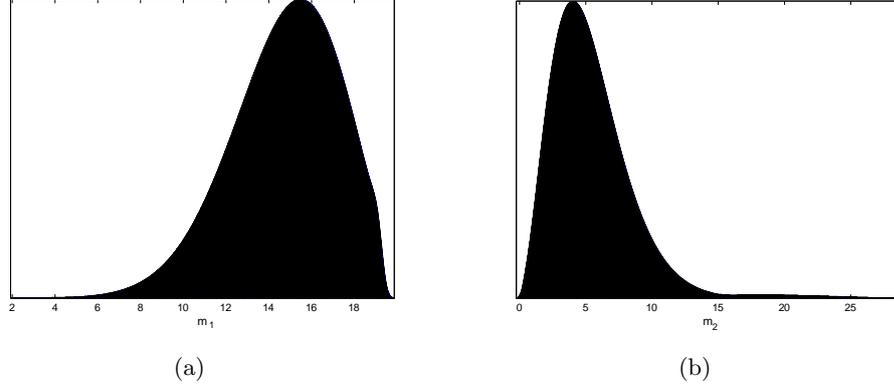
where  $m_1$  and  $m_2$  are the model parameters whose values must be estimated from the observed data.

The measurement errors are assumed to have reliable bounds  $[-\delta_i, \delta_i]$ ,  $1 \leq i \leq k$ , where  $\delta_i$  is an acceptable difference between the  $i^{th}$  observation  $y_i$  and the respective predicted value. Moreover observations are assumed to be more likely near the predicted values and this is modeled by a cosine distributions of the errors<sup>1</sup>  $\mathcal{C}(\delta_i, 0)$  such that the PDFs are

$$p_i(\varepsilon_i) = \frac{1}{2\delta_i} \left[ 1 + \cos \left( \frac{\varepsilon_i}{\delta_i} \pi \right) \right], \quad \varepsilon_i \in [-\delta_i, \delta_i]$$

---

<sup>1</sup>A Gaussian distribution was also used with similar results. Nevertheless this choice illustrates the generality of the framework.



**Figure 6.6:** Epicentral coordinates of the seismic event. Marginal distributions (a) for  $m_1$  and (b) for  $m_2$ .

The two parameters are assumed to have initial ranges  $I_1$  and  $I_2$  and to be uniformly distributed  $\mathcal{U}(\underline{I}_j, \overline{I}_j)$ , with  $1 \leq j \leq 2$ .

Assuming independence in the error measurements, the joint PDF for the model parameters and measurement errors is:

$$p_{M,E}(m_1, m_2, \varepsilon_1, \dots, \varepsilon_{13}) = \alpha_m \prod_{i=1}^{13} p_i(\varepsilon_i) \quad (6.8)$$

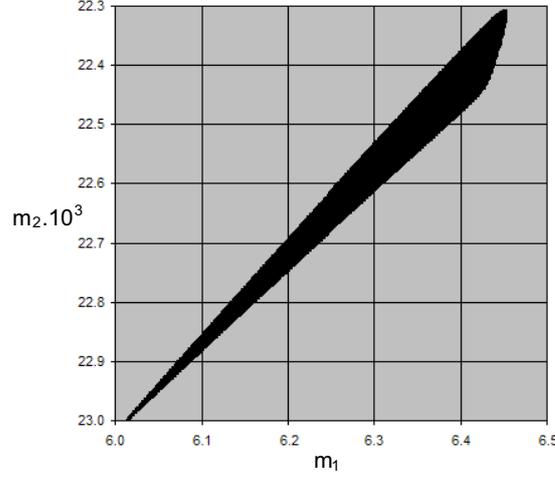
where  $\alpha_m$  is a constant resulting from the uniform parameter distributions.

Applying bounded-error estimation to this inverse problem, it can be formulated as a CCSP  $P = \langle X, D, C \rangle$  such that:

$$P = \langle \langle m_1, m_2 \rangle, D = I_1 \times I_2, \{y_i - \delta_i \leq m_1 e^{m_2 t_i} \leq y_i + \delta_i : 1 \leq i \leq 13\} \rangle \quad (6.9)$$

In the following consider  $D = [0, 100] \times [0.01, 0.1]$  and  $\delta_i = 3$  for all observations presented in Table 6.2.

Figure 6.7 shows an enclosure of the feasible space that is computed by *cReasoning* algorithm (algorithm 3) with arguments  $\langle \{D\}, \emptyset \rangle$ , the constraints of  $P$ , *stop*  $\equiv$  *false*, *order*  $\equiv$  *order*<sub>↓</sub> and the other parameters use their defaults with  $\varepsilon = 10^{-5}$  (see section A.1, appendix A). From the figure, it is clear which combinations of the model parameter values are consistent with the initial uncertainty assumptions, the forward model and the observations.



**Figure 6.7:** Enclosure of the CCSP feasible space.

This inverse problem may be formulated as a PC with:

$$\begin{aligned} \langle X, D, C \rangle &= P && \text{from (6.9)} \\ p(m_1, m_2) &= \alpha \prod_{i=1}^{13} p_i(y_i - m_1 e^{m_2 t_i}) && \text{from property 6.1 and (6.8)} \end{aligned}$$

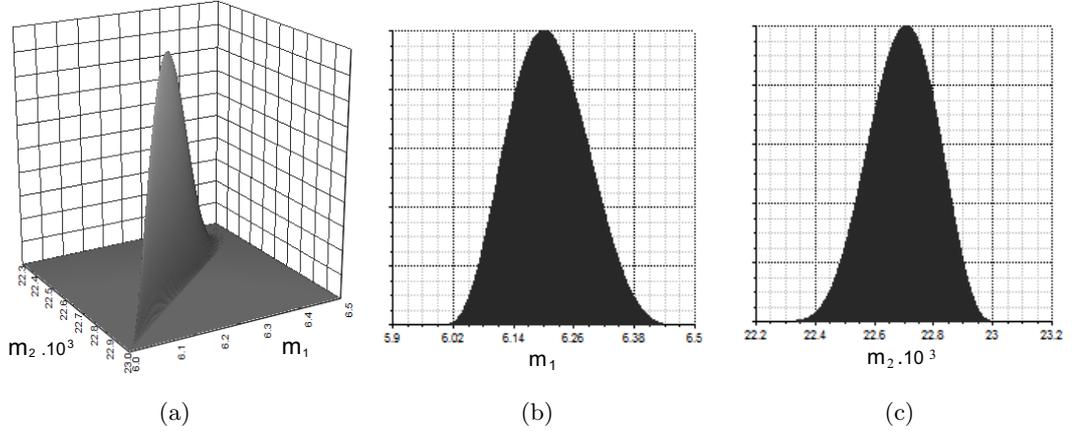
Figure 6.8 shows conditional distributions computed by algorithm 12 with  $\delta = 0.001$ , grid spacings  $\alpha = \langle 10^{-3}, 10^{-6} \rangle$  and  $\varepsilon = 10^{-15}$ . For figures 6.8 (a)  $\mathbf{Z} = \langle m_1, m_2 \rangle$ , (b)  $\mathbf{Z} = \langle m_0 \rangle$  and (c)  $\mathbf{Z} = \langle m_1 \rangle$  and so they present, respectively, the joint probability distribution of the model parameters and their marginal distributions.

Besides identifying which value combinations of  $m_1$  and  $m_2$  are consistent, figure 6.8 (a) illustrates their joint conditional probability distribution, allowing to identify regions of maximum likelihood.

If instead of an exponential model, a logistic model is considered for the population growth, then the forward model is defined by the set of equations:

$$y_i = f_i(t_i, m_1, m_2, m_3) = \frac{m_3}{1 + m_1 e^{-m_2 t_i}}$$

where  $m_1$ ,  $m_2$  and  $m_3$  are the model parameters, assumed to have initial ranges  $I_1$ ,  $I_2$  and  $I_3$  and to be uniformly distributed  $\mathcal{U}(\underline{I}_j, \overline{I}_j)$ , with  $1 \leq j \leq 3$ .



**Figure 6.8:** Exponential model. (a) Joint distribution; Marginal distributions (b) for  $m_1$  and (c) for  $m_2$ .

Assuming independence in the error measurements, the joint PDF for the model parameters and measurement errors is now:

$$p_{\mathbf{M}, \mathbf{E}}(m_1, m_2, m_3, \varepsilon_1, \dots, \varepsilon_{13}) = \alpha_m \prod_{i=1}^{13} p_i(\varepsilon_i) \quad (6.10)$$

where  $\alpha_m$  is a constant resulting from the uniform parameter distributions.

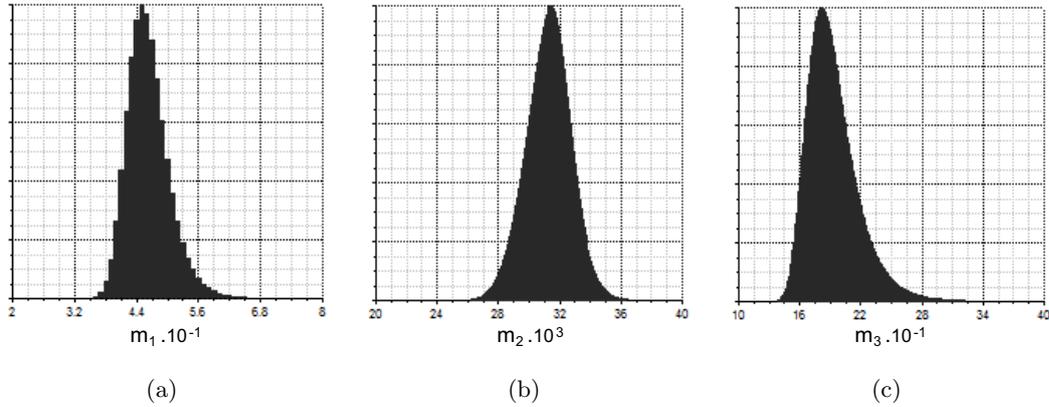
The formulation of this new inverse problem as a PC is given by:

$$\begin{aligned} X &= \langle m_1, m_2, m_3 \rangle \\ D &= I_1 \times I_2 \times I_3 \\ C &= \left\{ y_i - \delta_i \leq \frac{m_3}{1 + m_1 e^{-m_2 t_i}} \leq y_i + \delta_i : 1 \leq i \leq 13 \right\} \\ p(m_1, m_2, m_3) &= \alpha \prod_{i=1}^{13} p_i \left( y_i - \frac{m_3}{1 + m_1 e^{-m_2 t_i}} \right) \end{aligned}$$

Again we consider  $D = [10, 100] \times [0.02, 0.05] \times [100, 400]$  and  $\delta_i = 3$  for all observations presented in Table 6.2.

Figure 6.9 shows conditional distributions computed by algorithm 12 with  $\delta = 0.01$ , grid spacings  $\alpha = \langle 1, 10^{-4}, 1 \rangle$  and  $\varepsilon = 10^{-15}$ . For figures 6.8 (a)  $\mathbf{Z} = \langle m_1 \rangle$ , (b)  $\mathbf{Z} = \langle m_2 \rangle$  and (c)  $\mathbf{Z} = \langle m_3 \rangle$ .

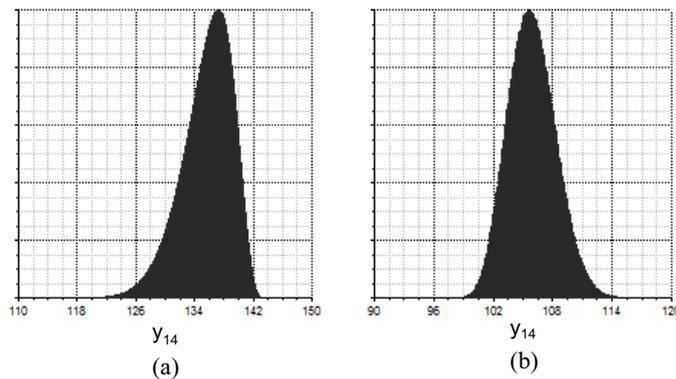
Figure 6.10 illustrates the predictions for the population size in 1920 ( $t_{14} = 130$ ) in



**Figure 6.9:** Logistic model. Marginal distributions (a) for  $m_1$ , (b) for  $m_2$  and (c) for  $m_3$ .

the previous problem with both, (a) the exponential model and (b) the logistic model. This information was computed with algorithm 13, using a parametrization similar with the ones described above (both for the exponential and logistic models) but with  $G = \{[g_{14}](\mathbf{m}) = m_1 e^{m_2 t_{14}}\}$  for the exponential model and  $G = \left\{ [g_{14}](\mathbf{m}) = \frac{m_3}{1 + m_1 e^{-m_2 t_{14}}} \right\}$ <sup>1</sup> for the logistic model.

Note that the real observed value for the population size in 1920 was 106.0 (not shown in table 6.2) which is in good agreement with the predictions of the logistic model, but outside the bounds predicted by the exponential model.



**Figure 6.10:** Expected US population in 1920. (a) Exponential and (b) logistic models.

An insight about the quality of a particular model for a specific inverse problem may

<sup>1</sup>In both cases,  $[g_{14}]$  is the natural inclusion function for  $g_{14}$ .

be achieved by analyzing the maximum likelihood regions. The marginal conditional distributions for the model parameters provide valuable information for inspecting the quality of a particular model. Not only they allow easy identification of maximum likelihood regions as peaks of such distributions, but also display the complete shape of the uncertainty dispersion showing, for instance, if it is unimodal.

In the presented example, given the unimodality of the conditional distributions for both models, a quantitative measure of their quality may be obtained by evaluating any numerical best-fit criterion at their maximum likelihood points. The boxes that enclose such points for the exponential and the logistic models are, respectively,  $\langle [6.159, 6.160], [0.022770, 0.022771] \rangle$  and  $\langle [45, 46], [0.0318, 0.0319], [181, 182] \rangle$ . The least squares criterion (formula (6.4)) evaluated at these boxes results, respectively, in  $I_1 = [7.78, 7.82]$  and  $I_2 = [0.39, 3.43]$ . Since the maximum likelihood points are included in those boxes and any value of  $I_2$  is smaller than any value of  $I_1$ , according to the chosen criterion, the logistic model is a better representation for the population growth than the exponential model.

## 6.6 Ocean Color Inversion

This section illustrates how the PC framework can be used in Ocean Color (OC), a research area which is widely used in climate change studies and has potential applications in water quality monitoring.

The aim of ocean color data inversion is to determine concentrations of optically active seawater compounds (OC products) of the water from observed remote sensing reflectance. Semi-analytical approaches [80, 85, 118] handle this problem as a nonlinear inverse problem where field data is used to configure a forward mathematical model that expresses sea-surface reflectance as a function of the OC products. Thus the PC framework can be applied to invert the forward model and compute all OC product scenarios consistent with the model, characterized by a probability distribution conditioned by the measurement error.

Such information is of extreme importance to understand the impact of measurement uncertainties on the derived OC products, providing support to: a) investigate the applicability of ocean color inversion schemes in different water types; and b) define

accuracy requirements for the radiometric sensors to guarantee specified levels of uncertainty for the estimated concentrations. This is an innovative contribution to the OC community, enabling a more informative exploitation of remote sensing products. Furthermore, this approach is general and can be extended to different parameterizations of the semi-analytical model.

### 6.6.1 Ocean Color

Ocean Color (OC) studies rely on the fact that, as sunlight enters the ocean, it interacts with particulate and the dissolved materials, besides the seawater itself. Inherent Optical Properties (IOPs) of the water quantify the result of this interaction in terms of scattering and absorption values. The light fraction that ultimately leaves the sea-surface and can be measured from space borne sensors (after correcting for the atmosphere contribution) is a function of these IOP values.

OC derived products include inorganic and organic optically active seawater compounds, that can change the magnitude and spectral characteristics of the radiance leaving the sea-surface by scattering and absorbing light in different ways. These compounds can be estimated based on the relation between them and the remote sensing measurements of sea-surface reflectance.

Semi-analytical methods are based on forward OC models (the analytical part) that express sea-surface reflectance as a function of optically active seawater compounds. The functional form of the forward model results from the radiative-transfer theory. However, some of the model coefficients rely on *in situ* bio-optical measurements (the empirical part).

Semi-analytical approaches mostly follow the same general forward model (varying on its configuration) and aim to estimate seawater compounds from satellite radiometric measures of sea-surface reflectance at specific wavelengths.

#### Forward Model

The forward relation between the remote sensing reflectance ( $R_{rs}$ ) at a given wavelength ( $\lambda$ ) and the IOPs (absorption  $a$ , and backscattering  $b_b$ ) is modeled as in [94] and shown in figure 6.11.

$$R_{rs}(\lambda) = 0.044 \times \frac{b_b(\lambda)}{a(\lambda) + b_b(\lambda)}$$

**Figure 6.11:** The forward model is a function from the optically active seawater compounds (*Chla*, *NPPM* and *CDOM*) to the remote sensing reflectance ( $R_{rs}$ ) at a given wavelength ( $\lambda$ ).

The OC products targeted in this study are the total concentration of chlorophyll-a and phaeopigments (*Chla*), the concentration of non-pigmented particles (*NPPM*) and the colored dissolved organic matter absorption at 400nm wavelength (*CDOM*). The total absorption,  $a$ , results from the additive contribution of the absorption of seawater ( $a_w$ ), phytoplankton ( $a_{Chla}$ ), non-pigmented particulate matter ( $a_{NPPM}$ ) and colored dissolved organic matter ( $a_{CDOM}$ ) at a given wavelength ( $\lambda$ ). The total backscattering,  $b_b$ , is given by the additive contribution of the backscattering of seawater ( $b_w$ ), phytoplankton ( $b_{Chla}$ ) and non-pigmented particulate matter ( $b_{NPPM}$ ) at a given wavelength ( $\lambda$ ). The colored dissolved organic matter does not produce any light scattering effect. The  $a_w(\lambda)$  and  $b_w(\lambda)$  are constant for a given wavelength. The  $A(\lambda)$  and  $B(\lambda)$  constants parameterize the phytoplankton chlorophyll-specific absorption coefficient.

### 6.6.2 Related Work

In many OC studies, the accuracy of the inverted parameters is characterized by means of standard statistical measures (e.g., mean relative error, root mean square error or coefficient of determination), by comparison with *in situ* measurements [3]. These procedures are dependent on the data set used to make the comparison, and are not obtained by uncertainty propagation through the mathematical model. Moreover this analysis cannot provide reliable estimates of how ocean color uncertainties vary with time and space, since they output a single global value for uncertainty characterization, based on the bulk of data available.

Some methods have been recently proposed to characterize uncertainty in a case-by-case basis (e.g., taking into account variations in time and space). In [85] the authors use a method for constructing the confidence intervals after nonlinear regression, based on linear approximations. This method was proposed in [11] and is also used in other OC approaches.

In [118], the original nonlinear model is transformed into a set of predefined linear model configurations which are inverted, resulting in a large set of possible solutions subsequently filtered to keep only those that closely (given an acceptance criteria) reproduce the input reflectance. The value of the parameters and their uncertainty estimate are given, respectively, by the median and percentiles obtained from the final set of solutions.

More recently, Lee et. al [80] proposed a method based on the algebraic inversion of the forward model to obtain a quasi-analytical algorithm from the remote sensing reflectance to the OC products. Based on the theory of error propagation over such algorithm, analytical expressions are derived to describe how the uncertainty on the remote sensing reflectance propagates into the OC products. However, the algebraic inversion relies on simplifications of the forward model and the results obtained based on the theory of error propagation may be compromised by the problem of nonlinearity.

The PC framework is suitable to characterize uncertainty in a case-by-case basis. Besides the innovative inversion method, the main difference to the existing methods is that it outputs the range of values, for the OC products, that are consistent with the given formulation of the model and the noisy measurements. This, *per se*, is already an add-on of this methodology, since existing similar approaches are reduced to output a single set of values for the retrieved OC products (the most likely scenario). Moreover, the proposed approach is able to fully propagate the uncertainty of the noisy measurements (expressed by their PDFs), through the mathematical model and produce probability distributions for the computed range of values.

### 6.6.3 Probabilistic Constraint Approach

Consider the inverse problem that estimates OC products from reflectance measurements, obtained at different wavelengths, based on the forward model of section 6.6.1.

$\lambda_i$	$K_1(\lambda_i)$	$K_2(\lambda_i)$	$K_3(\lambda_i)$	$K_4(\lambda_i)$	$K_5(\lambda_i)$	$K_6(\lambda_i)$	$K_7(\lambda_i)$	$K_8(\lambda_i)$
412	0.00544	0.03224	0.7130	0.05645	1.6096	0.00008564	0.007329	0.010832
443	0.00902	0.03935	0.6565	0.03855	0.9503	0.00006368	0.006816	0.010446
490	0.01850	0.02726	0.6384	0.02163	0.4274	0.00004209	0.006162	0.009933
510	0.03820	0.01804	0.7382	0.01691	0.3042	0.00003477	0.005921	0.009736
555	0.06900	0.00703	0.9693	0.00972	0.1416	0.00002562	0.005441	0.009333
670	0.43460	0.01848	0.8522	0.00236	0.0200	0.00001098	0.004507	0.008494

**Table 6.3:** Matrix of coefficients computed from the forward model of figure 6.11 and the values for the constants.

The model parameters are the OC products *Chla*, *CDOM* and *NPPM*, represented by variables  $m_1$ ,  $m_2$  and  $m_3$ . The observable parameters are all the reflectance measurements  $y_i$  obtained at wavelengths  $\lambda_i$ . Six different wavelengths are considered, corresponding to the central bandwidths of the SeaWiFS [86] sensor, represented by the vector  $\boldsymbol{\lambda} = (412, 443, 490, 510, 555, 670)$  nm.

Based on [121] we assume Gaussian measurement error distributions  $p_i$  with mean  $\mu_i = 0$  and where standard deviation,  $\sigma_i$ , is 5% of the obtained measurement  $y_i$  (except for  $\lambda_6$  in which it is 6% of  $y_6$ ):

$$p_i(\varepsilon_i) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{\varepsilon_i}{\sigma_i} \right)^2} \quad \text{with} \quad \sigma_i = \begin{cases} 0.05 \times y_i, & 1 \leq i \leq 5 \\ 0.06 \times y_i, & i = 6 \end{cases} \quad (6.11)$$

Given the above wavelengths and based on the forward model,  $R_{rs}(\lambda)$  (in figure 6.11) we define the mapping from the model parameters to the observed parameters, where  $1 \leq i \leq 6$ :

$$y_i = f(\lambda_i, m_1, m_2, m_3) = \frac{0.044}{\frac{K_1(\lambda_i) + K_2(\lambda_i)m_1^{K_3(\lambda_i)} + K_4(\lambda_i)m_2 + K_5(\lambda_i)m_3}{K_6(\lambda_i) + K_7(\lambda_i)m_1^{0.62} + K_8(\lambda_i)m_3} + 1} \quad (6.12)$$

and  $K$  is a  $6 \times 8$  matrix of coefficients (shown in table 6.3), calculated from the forward model of figure 6.11.

Assuming ranges between 0 and 50 for all model parameters and a maximum error of 3 standard deviations from the measured values, the inverse problem is modeled as a

PC  $\langle \langle m_1, m_2, m_3 \rangle, [0, 50]^3, C \rangle, p$  where:

$$C = \{y_i - 3\sigma_i \leq f(\lambda_i, m_1, m_2, m_3) \leq y_i + 3\sigma_i : 1 \leq i \leq 6\} \quad (6.13)$$

$$p(m_1, m_2, m_3) = \alpha \prod_{i=1}^6 p_i(y_i - f(\lambda_i, m_1, m_2, m_3)) \quad (6.14)$$

### 6.6.4 Experimental Results

This section shows the kind of results the PC framework is able to provide, its innovative contributions and the limitations of alternative available methods. The work is not meant to make an expert critical analysis on the parameter values and uncertainty results that were obtained.

All experiments were carried out on an Intel Core i7 CPU at 1.6 GHz and are based on any time implementations of algorithms 11 (page 130) and 12 (page 134)<sup>1</sup>. Two versions of the algorithms were considered in the experiments: PCTM uses the validated quadrature method based on Taylor models (with order 4) and provides safe enclosures of the computed quantities; PCMC uses Monte Carlo integration and provides estimates for the computed quantities.

In this context we studied a set of 12 simulated cases representative of different seawater types that can be found in nature. For each, the model parameters were established (see table 6.4) and the values of the observed parameters generated with the forward model. Such simulated values were used instead of real measurements so that the estimated parameter values can be compared with the exact values used to simulate the observations.

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
<i>Chla</i>	1	5	10	0	0	0	0	0	0	0.1	1	10
<i>NPPM</i>	0.374	1.141	1.843	0.5	1	5	0.5	0.5	0.5	0.1	0.5	2
<i>CDOM</i>	0.012	0.035	0.055	0	0	0	0.1	0.5	2	0.5	5	0.5

**Table 6.4:** The 12 experimental cases were simulated by using the forward model to compute the observed parameter values from the above model parameters values.

<sup>1</sup>This is achieved by replacing the stopping criterion of the *cReasoning* function call at line 7 of algorithm 12 with the adequate *stop<sub>T</sub>* predicate to impose the required runtime limit.

The results shown next are supported by experiments in all 12 cases. To simplify the interpretation, we present in more detail the results for a single case (#2), where the simulated observed values computed by equation (6.12) were:

$$\mathbf{y} = \mathbf{f}(\boldsymbol{\lambda}, 5, 1.141, 0.035) = (5.469, 5.831, 7.611, 7.746, 7.775, 1.811) \times 10^{-3}$$

### Uncertainty Characterization with Probabilistic Constraints

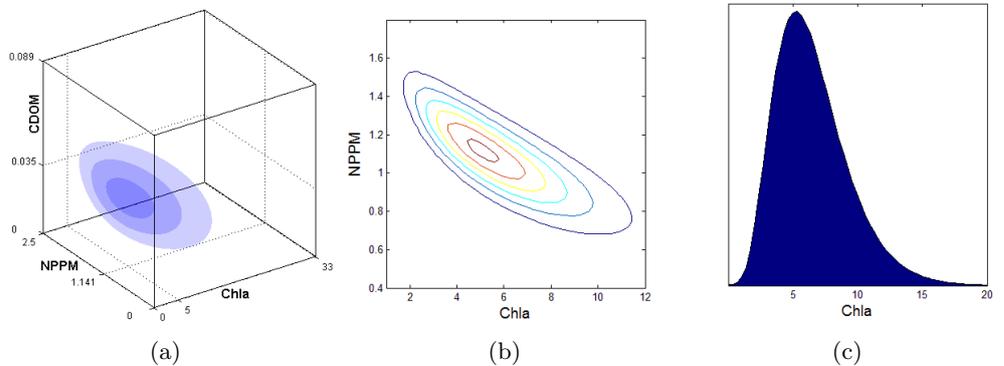
To characterize the uncertainty on the model parameters given the measurements, algorithm 12 is used over the conditioning event  $\mathcal{H} = \mathcal{F}(\langle\langle m_1, m_2, m_3 \rangle, [0, 50]^3, C \rangle)$ , with  $L = \langle 100, 100, 100 \rangle$ , imposing a grid defined by partitions of size 100 in each dimension. Conditional expected values  $E[m_i|\mathcal{H}]$  and variances  $Var[m_i|\mathcal{H}]$  for each model parameter are computed using the resulting joint box cover and algorithm 11.

In case (#2), algorithm 12 returns the bounding box  $H = [0, 32.9136] \times [0, 2.5025] \times [0, 0.0938]$  which encloses all possible values for the parameters *Chla*, *NPPM* and *CDOM*. This is not too informative, specially for the large range of consistent values for *Chla*. However, by considering a grid over this cover, the distribution of the uncertainty between the different consistent scenarios is quantified.

The results obtained with the anytime implementation of the PCTM version to compute those conditional probabilities together with the conditional expected values and variances for the model parameters are shown in figure 6.12.

The graphics of the figure are based on the midpoints of the interval enclosures for the grid boxes conditional probabilities, obtained by the algorithm after about 20 minutes of CPU time. Figure 6.12 (a) illustrates the joint distribution, which identifies regions of maximum likelihood (darker colors represent more likely regions). Figure 6.12 (b) shows the marginal distributions for the combination of *Chla* and *NPPM* values and figure 6.12 (c) the marginal distribution for *Chla*. Although not shown in the figures the computed grid box with the highest probability value  $[4.9455, 5.2746] \times [1.1206, 1.1455] \times [0.0345, 0.0354]$  includes the exact model parameter values (5, 1.141, 0.035) used to simulate the measured values.

Table 6.5 shows the results obtained by algorithm 11 to compute enclosures for the expected value of *Chla* ( $E[Chla|\mathcal{H}]$ ) and its standard deviation ( $\sqrt{Var[Chla|\mathcal{H}]}$ ). These



**Figure 6.12:** Joint and marginal uncertainty distributions computed by the PC framework.

enclosures, shown in columns 2 and 5 (with 4 digits precision), get sharper with time (column 1). Columns 3 and 6 show the enclosures midpoint and columns 4 and 7 its maximum error (half of the interval width).

	$E[Chla]$			$STD[Chla]$		
	enclosure	midpoint	error	enclosure	midpoint	error
10 min	[6.2339, 7.0518]	6.6429	0.4090	[2.3757, 3.1078]	2.7418	0.3661
20 min	[6.4522, 6.8483]	6.6503	0.1981	[2.5802, 3.0024]	2.7913	0.2111
60 min	[6.5428, 6.7583]	6.6505	0.1077	[2.6939, 2.9412]	2.8175	0.1236
300 min	[6.5893, 6.7119]	6.6506	0.0613	[2.7584, 2.9087]	2.8335	0.0751

**Table 6.5:** Interval enclosures of  $E[Chla]$  and  $STD[Chla]$  computed by the PCTM version of algorithm 11.

When we replace the Taylor model validated method with the Monte Carlo approximated method (PCMC version), results can be obtained much faster but without guarantees of correctness.

Table 6.6 shows the results obtained with the PCMC version of algorithm 12. Due to the added value of constraint programming to reduce the sampling space, only a small number of samples,  $N$ , within each grid box, is necessary to obtain approximations close to the correct values<sup>1</sup>. Furthermore, the graphics produced with this method are all very similar to the graphics presented in figure 6.12.

<sup>1</sup>Of course, such claim is only supported by comparison with the validated enclosures of table 6.5.

	$N$	$E[Chla]$	$STD[Chla]$
2 min	5	6.6625	2.8746
3 min	10	6.6623	2.8742
4 min	20	6.6624	2.8740
9 min	50	6.6623	2.8739

**Table 6.6:** Approximations of  $E[Chla]$  and  $STD[Chla]$  computed by the PCMC version of algorithm 11.

### The need for Constraint Programming

Notice that the PCMC version is a hybrid algorithm that benefits from the contribution of constraint programming to reduce the sample space into a sharp enclosure of the feasible space, combined with the efficiency of Monte Carlo integration to obtain fast approximate estimates for the parameters distribution.

These results could not be obtained as efficiently with a pure Monte Carlo approach (without the contribution of constraint programming). To test this assertion a pure adaptive Monte Carlo integration algorithm, adapted from [6], was implemented - denoted hereafter as AMC. The algorithm follows a stratified sampling technique to ensure that important regions get more samples. It applies a global subdivision strategy for partitioning the initial sample space  $D$  into sub-boxes, performing a basic Monte Carlo integration with  $N$  sampling points in each box to compute estimates of the integral and the standard error as presented in section 3.5.3. The algorithm keeps splitting the box with largest estimated standard deviation. The overall estimated integral, is kept updated as the sum of the estimated integrals of each box of the partition.

Table 6.7 shows the results obtained after 10 minutes CPU time by the AMC algorithm with several different sampling sizes  $N$ . Columns 2 and 3 are the estimates for  $Chla$  expected value and standard deviation. Column 4 shows the number of boxes in the partition when execution ends. Column 5 is the total number of solutions found during the sampling process. Column 6 presents an interval whose lower (upper) bound is the smallest (largest) value of  $Chla$  found in the sampled solutions.

One major difficulty for the application of the adaptive Monte Carlo algorithm to these studies is its high sensitivity to the sampling size  $N$ . As illustrated in table 6.7, with similar execution times and total number of samplings, when  $N$  increases,

$N$	$E[Chla]$	$STD[Chla]$	$\#boxes$	$\#solutions$	$[Chla]$
20	14.4406	0.2249	188102	6475002	[3.5154, 18.3783]
100	19.8885	0.8382	37785	6504800	[11.2931, 21.8806]
1000	14.2240	1.4436	4035	5845064	[12.5075, 18.7562]
10000	6.6492	2.8464	585	2579803	[0.0289, 24.9730]
100000	6.6837	2.8704	69	404389	[0.0455, 26.9990]
1000000	6.6863	3.0763	8	1692	[0.1778, 30.1990]
10000000	7.9807	3.1657	2	283	[0.4448, 30.8870]

**Table 6.7:** Approximations of  $E[Chla]$  and  $STD[Chla]$  computed by the AMC algorithm after 10 minutes with several sampling sizes  $N$ .

both the number of boxes in the final partition ( $\#boxes$ ) and the number of solutions found ( $\#solutions$ ), decrease.  $\#boxes$  decreases because each box takes longer to sample.  $\#solutions$  decreases because with less boxes the stratification effect is attenuated (stratification induces more sampling in regions where solutions have been previously found since their standard error estimations are larger).

With very large sampling sizes, the decrease of  $\#solutions$  prevents obtaining good approximations within reasonable execution times because the set of solutions found is not representative of the total integration region. On the other hand, with too small sampling sizes, the sampling is strongly biased towards regions of previously found solutions, and boxes where no solutions were found will not be further sampled. Therefore, with too small sampling sizes, despite its larger number, the set of solutions found is still not representative of the total integration region because stratification prevented the sampling of some regions (this is illustrated in the last column of table 6.7 where the  $Chla$  sampling ranges for small  $N$  are clearly not representative of all the possible  $Chla$  values - see figure 6.12 (c)).

In fact, good approximations for  $E[Chla]$  and  $STD[Chla]$ , can only be provided (within reasonable execution times) with an appropriate sampling size which is highly dependent on the feasible space. In table 6.7 it is clear that the best  $N$  value should be between 10000 and 100000. However, if the measurement accuracy is improved (as in the examples of table 6.8), these numbers become completely inadequate as the algorithm is no longer able to find enough solutions (with  $N = 10000$  no solution was found, in the 10% case, after 1 hour CPU time!).

In the present study, besides computing the overall expectations and standard deviations, we are interested in providing an overview of the uncertainty distribution. This is done by producing results as those shown in figure 6.12, in which a partitioning of the sample space with a good granularity around the feasible space is necessary. This is a major difficulty for the adaptive Monte Carlo algorithm, since, without constraint programming, such cannot be done in reasonable CPU time.

For example, the graphics in figure 6.12 were produced partitioning the enclosing box of figure 6.12 (a) into  $100 \times 100 \times 100 = 1000000$  grid boxes. Through constraint programming 803852 (over 80%) of those grid boxes were discarded (because they do not contain solutions) and only the remaining 196148 were used for the computation of their conditional probabilities. The complete process took about 2 minutes CPU time. From table 6.7 we can see that this cannot be achieved with the adaptive Monte Carlo algorithm. A similar granularity is only attainable with very small sampling sizes leading to the undesirable properties that we have discussed before.

### The Impact of Measurements Accuracy

An important contribution of the PC approach to the OC community is its ability to address different assumptions on the measurements accuracy to understand how these may affect the uncertainty on the retrieved OC products. Such studies can be used to define accuracy requirements for the radiometric sensors to guarantee specified levels of uncertainty for the estimated concentrations.

Table 6.8 shows the results obtained with different measurement accuracies. The first row was computed with the error standard deviation  $\sigma_i$  specified in equation (6.11) and the others with a percentage of that value (column 1). For each of these accuracies columns 2 and 4 show the enclosures for  $E[Chla]$  and  $STD[Chla]$  obtained by the PCTM version of algorithm 11 after 20 minutes of CPU time. The approximate values computed by the PCMC version of algorithm 11 are given in columns 3 and 5 respectively. All the PCMC computations were performed with a sampling size  $N = 5$  and took less than 1.5 minutes of CPU time (except the first one that took 2 minutes).

It is clear from table 6.8 that, with improved measurement accuracy, the expected value of  $Chla$  (similar results were obtained for the other OC products) converges to

$\sigma_i$	$E[Chla]$		$STD[Chla]$	
	PCTM	PCMC	PCTM	PCMC
100%	[6.4522, 6.8483]	6.6625	[2.5802, 3.0024]	2.8746
50%	[5.3235, 5.4604]	5.3949	[1.1343, 1.2653]	1.2293
10%	[4.9788, 5.0559]	5.0175	[0.1940, 0.2392]	0.2324
5%	[4.9789, 5.0335]	5.0063	[0.0921, 0.1198]	0.1163
1%	[4.9742, 5.0309]	5.0025	[0.0064, 0.0238]	0.0232

**Table 6.8:**  $E[Chla]$  and  $STD[Chla]$  obtained by the PC algorithms for different accuracies.

the exact value used to simulate the observations ( $Chla = 5$ ) and the standard deviation approaches zero. This provides insight on the magnitude of the incurred errors, with different sensor accuracies, that allows to estimate the OC product by its expected value, or justify the use of other estimates (e.g. the most likely value, i.e. that in the most probable grid box).

## 6.7 Summary

This chapter illustrated the application of the probabilistic continuous constraint framework to decision problems on nonlinear inverse problems. Inverse problems were defined and classical techniques to solve them were presented, highlighting drawbacks of such approaches. The definition of an inverse problem as a probabilistic continuous constraint space was presented and the capabilities of the PC framework were illustrated in three application problems. The first two showed how to deal with nonlinear inverse problems, in general. The last, more complex, problem showed that the framework can be used in real world applications.

The next chapter, following a similar structure, illustrates the application of the framework to reliability problems.

## CHAPTER 7

# Reliability Problems

Reliability analysis studies the ability of a system to perform its required function under variable conditions. In this context reliability assessment quantifies the chance of system failures at any stage of a system's life; reliability based design is concerned with choosing design alternatives that improve reliability, minimizing the risk of failure; and reliability based design optimization considers other criteria (e.g. cost minimization) in addition to maximizing reliability. This research area has application in a wide range of different industries including the aeronautical [99], nuclear [78], chemical [47] and building [65] industries.

When modeling a design problem there is often a distinction between controllable (or design) variables, representing alternative actions available to decision makers, and uncontrollable variables (or states of nature) corresponding to external factors outside their reach. Uncertainty affects both types of variables. There can be variability on the actual values of the design variables (e.g. the exact intended values of physical dimensions or material properties may not be obtained due to limitations of the manufacturing process). Or there can be uncertainty due to external factors that represent states of nature (e.g. earthquakes, wind). In both cases, it is important to quantify the reliability of a chosen design.

Reliability is often reported in terms of the probability of adequate functioning of a system and its exact quantification requires the calculation of a multi-dimensional integral with a non-linear integration boundary. Because there is rarely a close-form solution, this calculation is one of the major concerns of classical approaches to solve reliability problems, which adopt approximation methods that rely on several simplifications

of the original problem to compute a reliability estimate, often leading to inaccurate results, especially in highly non-linear problems.

Since designs with a high reliability estimate but high uncertainty on such estimation are not credible solutions, it is important to obtain bounds to such estimate. In practice, decision makers prefer an option with a marginally lower reliability estimate but where safe lower bounds to this estimate are computed with sound engineering models and techniques.

Such safe bounds are not available with classical approaches, but they can be provided by constraint programming which focuses on finding values for design variables that satisfy the constraints of a problem, since these are the variables over which the problem solver has some degree of choice. However, decisions have to be made taking into account the uncertainty regarding the uncontrollable variables.

A possible approach to deal with such uncertainty is to adopt a quantified constraint paradigm [13], where uncontrollable variables are assumed to be universally quantified, and the goal is to find values of design variables that satisfy the problem constraints, for any possible values of the uncontrollable variables. However, not all scenarios are equally likely, and this safe approach is often inadequate, as it does not provide solutions with a high likelihood to succeed.

In this chapter we show the ability of the Probabilistic Continuous Constraint framework to model reliability problems and thus compute safe bounds for the reliability of a system, allowing to distinguish between different system design scenarios and address reliability based design optimization. Section 7.1 introduces the concepts of reliability assessment, reliability based design and reliability based design optimization. Section 7.2 describes the classical techniques used to address those problems, pointing their main drawbacks. In section 7.3 the probabilistic constraint approach to address the problem formulations of reliability analysis is presented and the capabilities of the PC framework are highlighted in a set of illustrative examples. In section 7.4 a common engineering benchmark is studied (short rectangular column), where the advantages of using the PC framework are highlighted.

## 7.1 Reliability Analysis

A limit-state is a condition beyond which a system no longer fulfills the desired functionality. Reliability analysis calculates and predicts the probability of limit-state violations at any stage of a system's life.

The probability of occurrence of a limit-state violation in a system represents its probability of failure,  $P_f$ , whereas  $P_s = 1 - P_f$  represents its reliability. Failure events are represented as limit-state constraints:

$$g(\mathbf{x}) < 0 \quad (7.1)$$

where  $g$  is a limit-state function and  $\mathbf{x}$  is a realization of the random vector  $\mathbf{X}$  (defined in  $\Omega_{\mathbf{X}}$ ) that represents all the relevant uncertainties influencing the probability of failure and has joint PDF  $f_{\mathbf{X}} : \Omega_{\mathbf{X}} \rightarrow [0, +\infty]$ . Whereas  $g(\mathbf{x}) < 0$  denotes the failure region,  $g(\mathbf{x}) = 0$  and  $g(\mathbf{x}) > 0$  indicate the failure surface and safe region, respectively. As such, the failure and success events are:

$$F = \{\mathbf{x} \in \Omega_{\mathbf{X}} : g(\mathbf{x}) < 0\} \quad \text{and} \quad S = \{\mathbf{x} \in \Omega_{\mathbf{X}} : g(\mathbf{x}) \geq 0\}$$

So the probability of failure,  $P_f$ , is defined as the probability of the failure event:

$$P_f = P(\mathbf{x} \in F) = \int_{g(\mathbf{x}) < 0} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad (7.2)$$

Consequently, the probability of success,  $P_s$ , i.e. the reliability, is its complement:

$$P_s = P(\mathbf{x} \in S) = \int_{g(\mathbf{x}) \geq 0} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \quad (7.3)$$

In fact, we are interested in problems that can be defined by one or more limit-state functions, as the one in equation (7.1). In this context, in a series system, its global failure occurs when at least one of various limit-state functions is violated. Whereas, in a parallel system, its global failure occurs when all the limit-state functions are violated. So, for a series system with  $n$  limit-state functions,  $g_i$ , the failure and success events

are:

$$F^S = \bigcup_{i=1}^n \{\mathbf{x} \in \Omega_{\mathbf{X}} : g_i(\mathbf{x}) < 0\} \quad \text{and} \quad S^S = \bigcap_{i=1}^n \{\mathbf{x} \in \Omega_{\mathbf{X}} : g_i(\mathbf{x}) \geq 0\} \quad (7.4)$$

Thus, its probability of failure and reliability are, respectively,  $P_f^S = P(\mathbf{x} \in F^S)$  and  $P_s^S = P(\mathbf{x} \in S^S)$ .

Likewise, for a parallel system with  $n$  limit-state functions, the failure and success events are:

$$F^P = \bigcap_{i=1}^n \{\mathbf{x} \in \Omega_{\mathbf{X}} : g_i(\mathbf{x}) < 0\} \quad \text{and} \quad S^P = \bigcup_{i=1}^n \{\mathbf{x} \in \Omega_{\mathbf{X}} : g_i(\mathbf{x}) \geq 0\} \quad (7.5)$$

And its probability of failure and reliability are, respectively,  $P_f^P = P(\mathbf{x} \in F^P)$  and  $P_s^P = P(\mathbf{x} \in S^P)$ .

### 7.1.1 Reliability Based Design

Contrary to conventional deterministic design, where generally empirically based safety factors are used, reliability based design (RBD) directly relates design parameters to the reliability, or its complement, the probability of failure.

The basic goal of RBD is to ensure that the probability of failure of a system does not exceed an acceptable threshold, and an acceptable design (i.e., a value assignment to the design parameters) would be one in which the probability of failure respects this threshold.

In this context, the limit-state function depends also on the design parameters  $\mathbf{Y}$  and is defined as  $g(\mathbf{x}, \mathbf{y})$  where  $\mathbf{y} \in \Omega_{\mathbf{Y}}$ . As such, the failure and success events associated to each decision  $\mathbf{y} \in \Omega_{\mathbf{Y}}$  are<sup>1</sup>:

$$F(\mathbf{y}) = \{\mathbf{x} \in \Omega_{\mathbf{X}} : g(\mathbf{x}, \mathbf{y}) < 0\} \quad \text{and} \quad S(\mathbf{y}) = \{\mathbf{x} \in \Omega_{\mathbf{X}} : g(\mathbf{x}, \mathbf{y}) \geq 0\}$$

Thus, the RBD goal can be stated as the constraint:

$$P_f = P(\mathbf{x} \in F(\mathbf{y})) \leq tol \quad (7.6)$$

---

<sup>1</sup>When series or parallel systems are considered these events can be adapted based on, respectively, (7.4) and (7.5).

in which  $tol$  is an acceptable target probability of failure. As such, reliability based design, assumes a conscious choice on an acceptable level of design risk and then proceeds to a particular design consistent with this choice. In contrast to the traditional or partial factors of safety approach, consistency between the computed design risk and the uncertainties inherent in the design process is assured by reliability analysis.

In practice many reliability based design problems include optimization criteria and are called reliability based design optimization (RBDO) problems [38]. Besides the information about the failure mode of a system (modeled by the limit-state functions), they include information about its desired behavior, modeled by one or more objective functions,  $h_i$ , over the uncontrollable variables  $\mathbf{X}$  and design parameters  $\mathbf{Y}$ . The aim is to obtain reliable decisions that are optimal wrt the objective functions.

Formally, a reliability based optimization design problem can be stated as:

$$\begin{aligned} & \text{Minimize} && h_i(\mathbf{x}, \mathbf{y}) \\ & \text{subject to} && P(\mathbf{x} \in F(\mathbf{y})) \leq tol, \\ & && \mathbf{x} \in \Omega_{\mathbf{X}}, \mathbf{y} \in \Omega_{\mathbf{Y}}. \end{aligned}$$

Within this formulation of an RBDO problem, several cases can be considered [38]. For instance, if we want to minimize the probability of failure, one of the objective functions is  $P_f$ . In multi objective RBDO, instead of a single optimal solution, a Pareto-optimal frontier may be obtained to represent the best solutions.

In this context, a Pareto-optimal frontier is the set of designs not strictly dominated by another design. Consider two designs  $\mathbf{y}_1$  and  $\mathbf{y}_2$ , then  $\mathbf{y}_1$  strictly dominates  $\mathbf{y}_2$  if it satisfies the Pareto criterion:

$$\forall_i h'_i(\mathbf{y}_1) \leq h'_i(\mathbf{y}_2) \bigwedge \exists_i h'_i(\mathbf{y}_1) < h'_i(\mathbf{y}_2)$$

where  $h'_i(\mathbf{y})$  is defined from a suitable transformation of  $h_i(\mathbf{x}, \mathbf{y})$ <sup>1</sup>.

---

<sup>1</sup>One possibility is to use the conditional expected value,  $h'_i(\mathbf{y}) = E[h_i(\mathbf{x}, \mathbf{y}) | g(\mathbf{x}, \mathbf{y}) \geq 0]$ .

## 7.2 Classical Techniques

Reliability assessment involves the calculation of a multi dimensional integral in a possibly highly non-linear integration boundary (equations (7.2) or (7.3)). Analytical computation of such integral is usually impossible, so various simulation-based and analytical methods have been proposed to deal with this problem.

In [60], Hasofer and Lind introduced the reliability index technique for calculating approximations of the desired integral with reduced computation costs. The reliability index has been extensively used in the first and second order reliability methods (FORM [64] and SORM [46]). Using such methods, an approximation to  $P_f$  can be obtained by analytical techniques.

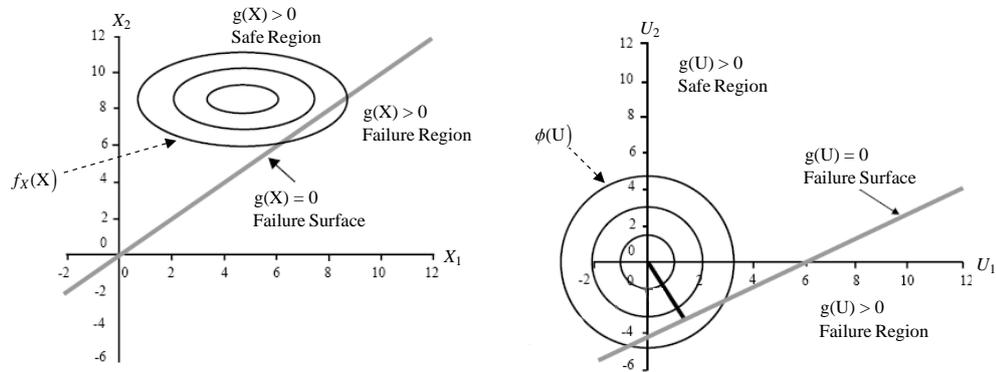
The main idea is to move the reliability problem from the space of random vector  $\mathbf{X}$  to the space of standard normal statistically independent random variables  $\mathbf{U} = \langle U_1, \dots, U_n \rangle$  using a suitable transformation  $\mathbf{U} = T(\mathbf{X})$ , such as Roseblatt [106] or Nataf [95] transformations (see [63, 87] for an overview). In the  $\mathbf{U}$  space, equation (7.2) can be expressed as:

$$P_f = \int_{g(\mathbf{u}) \leq 0} f_{\mathbf{U}}(\mathbf{u}) d\mathbf{u} = \prod_{i=1}^n \int_{g(\mathbf{u}) \leq 0} \phi_{U_i}(u_i) du_i$$

where  $\phi_{U_i}$  is the standard normal PDF of random variable  $U_i$ .

This process is illustrated in figure 7.1 in a two dimensional case. Figure 7.1 (a) shows the original  $\mathbf{X}$  space, with random vector  $\mathbf{X} = \langle X_1, X_2 \rangle$ . For illustration purposes we assume independent random variables  $X_1 \sim \mathcal{N}(\mu_1, \sigma_1)$  and  $X_2 \sim \mathcal{N}(\mu_2, \sigma_2)$ . Figure 7.1 (b) presents the standard normal  $\mathbf{U}$  space obtained using transformation  $\langle U_1, U_2 \rangle = T(\langle X_1, X_2 \rangle) = \langle \frac{X_1 - \mu_1}{\sigma_1}, \frac{X_2 - \mu_2}{\sigma_2} \rangle$ . The linear limit-state function  $g(\mathbf{x}) = ax_1 + bx_2 + c$  has been transformed into the limit-state function  $g(\mathbf{u}) = a\sigma_1 u_1 + b\sigma_2 u_2 + a\mu_1 + b\mu_2 + c$  by replacing  $x_i$  with  $\sigma_i u_i + \mu_i$ . Notice that, in this particular case, the transformed limit-state function is still linear but, in general, such transformations may induce non linearity in the resulting limit-state functions.

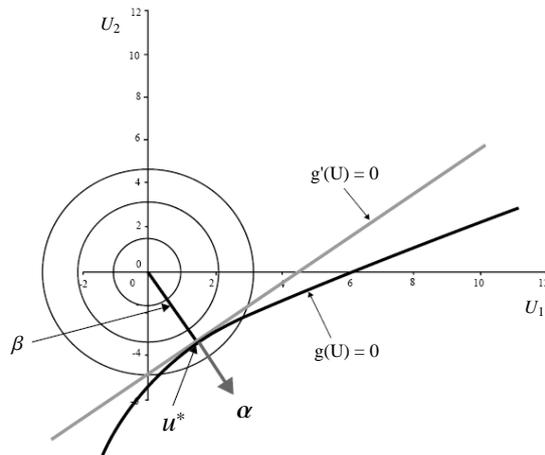
In FORM an approximation to the probability of failure is obtained by making the failure surface  $g(\mathbf{U}) = 0$  linear at the design point,  $\mathbf{u}^*$ , often called most probable



(a)  $X$  space where  $X$  is distributed as  $f_X$ . (b)  $U$  space where each  $U_i$  is standard normally distributed.

**Figure 7.1:**  $X$  space and  $U$  space.

point of failure (MPP). This is the point on the failure surface closest to the origin and with the highest probability (local maximum) in the failure domain of the standard normal space. The distance from the origin to the design point is the reliability index  $\beta = \|\mathbf{u}^*\|$ . These concepts are geometrically illustrated in figure 7.2.



**Figure 7.2:** Geometrical illustration of the reliability index.

Since the standard normal space is rotational symmetric the probability of failure can be directly obtained using the reliability index:

$$P_f = \Phi(-\beta) \tag{7.7}$$

where  $\Phi$  is the standard normal cumulative probability function.

As the limit state function is in general non-linear it is not possible to know the design point in advance and this has to be found iteratively. The design point is thus, the solution to the constrained optimization problem:

$$\beta = \min_{\mathbf{u} \in \{g(\mathbf{u})=0\}} \|\mathbf{u}\|$$

This problem, being the most expensive part of the FORM algorithm, may be solved in a number of different ways (see [42] for an overview). An appropriate iteration scheme converges after some iterations, providing the design point  $\mathbf{u}^*$  as well as the reliability index  $\beta$ , which may be related directly to the probability of failure as in equation (7.7). However, as with any non convex optimization problem, it is not guaranteed that the solution point will be the global minimum-distance point.

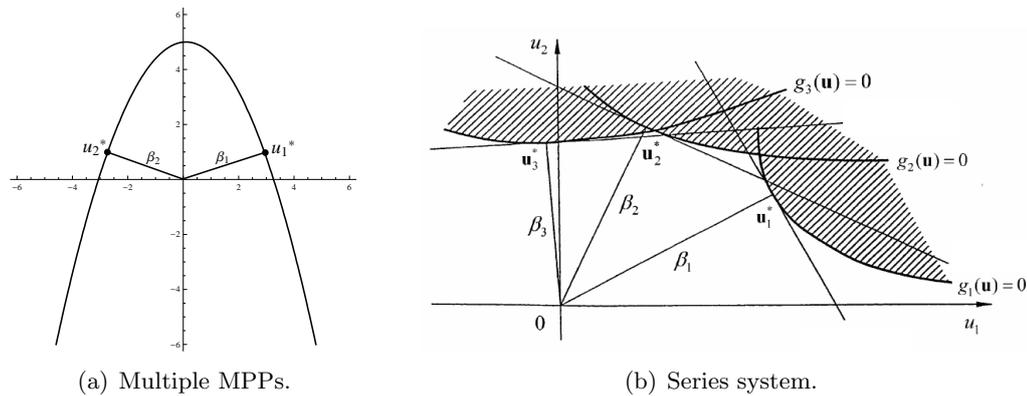
FORM usually works well when the failure surface has only one minimal distance point and the function is nearly linear in the MPP neighborhood. However, for increasingly non linear failure surface the probability of failure estimated by FORM becomes increasingly inaccurate (and possibly unreasonable) [87]. To address such non linearity SORM incorporates some curvature in the limit state approximation. In this case the probability of failure, based on the reliability index and on a correction factor, becomes [20]:

$$P_f = \Phi(-\beta) \prod_{i=1}^{n-1} \frac{1}{\sqrt{1 + \beta\kappa_i}}$$

where  $\kappa_i$  are the principal curvatures of the limit state. This method essentially uses a parabolic approximation to the failure surface being more accurate for large values of  $\beta$ .

In the methods discussed so far, it is assumed a single limit-state function with a single design point where only the region around such point contributes to the probability of failure. In limit-state functions with multiple design points (and in problems with multiple limit-state functions), application of FORM or SORM around a single design point results in erroneous estimates for the probability of failure. So, the problem of identifying the multiple design points must be addressed. In [73] a method is presented

to search for multiple design points: when one design point is identified, the failure surface is deformed around such point to avoid its repeated identification and the process is re-initialized to find the next design point until no more design points exist. In [10] the authors use a method based on evolutionary strategies to search for multiple design points. In series systems, generally there exists one design point for each limit-state function (which contributes to the identification of the feasible region). In contrast, in parallel systems there usually exists one design point for each pairwise intersection of limit-state functions (again contributing to the identification of the unfeasible region). Problems with multiple design points are illustrated in figure 7.3, (figure 7.3 (b) was adapted from [111]).



**Figure 7.3:** Problems with multiple design points.

Once all design points are identified, FORM or SORM approximations are constructed at these points and the failure probability is computed by series system reliability analysis (for multiple design points in a single limit-state function or series systems) or by parallel system reliability analysis (for parallel systems) (see [111, Notes 6 and 7] for details).

As already stated, reliability based design optimization intends to provide reliable designs wrt a set of objective functions. As such, the methods described above are used within the classical techniques that address such problems, including double-loop approaches [79] (consisting of a design optimization loop which repeatedly calls a reliability analysis method in a series of inner loops). The computational effort of such approaches may be prohibitive in some cases and other alternatives were proposed.

The sequential optimization and reliability assessment (SORA) [40] decouples the process in two sequential steps: a deterministic design optimization step followed by a set of reliability assessment loops. Other methods convert the problem into single-loop deterministic optimization [5, 31, 82]. For an overview of such methods see [38].

In general, the accuracy of the approximations computed with FORM and SORM is penalized by several assumptions taken to implement them, more noticeable when multiple design points exist.

A first assumption is that the joint PDF in (7.2) can be approximated by a multivariate Gaussian. Various normal transformation techniques must be applied [63] when the original space includes non-normal random variables which may lead to major errors.

A second assumption is that the feasible space determined by a single constraint can be reasonably approximated on the most probable point, on the constraint boundary. Instead of the original constraint, a tangent plane (FORM) or a quadratic surface (SORM), fitted at the MPP, is used to approximate the feasible region. However, the non linearity of the constraint may lead to unreasonable approximation errors. Firstly, local optimization methods [60] used to search for the MPP are not guaranteed to converge to a global minimum. Secondly, an approximation based only on a single MPP does not account for the possibly significant contributions from the other points [73]. Finally, the linear or quadratic approximation of the constraint may be unrealistic for highly non-linear constraints.

A third assumption is that the overall reliability can be reasonably approximated from the individual contributions of each constraint when a series system is considered. In its simplest form, only the most critical constraint is used to delimit the unfeasible region. This may obviously lead to over estimation of the overall reliability. More accurate approaches [39] take into account the contribution of all the constraints but, to avoid overlapping the contribution of each pair of constraints, they have to rely on approximations of the corresponding joint bivariate normal distributions.

Sampling techniques, based on Monte Carlo simulation (MCS) [55], work well for small reliability requirements, but as the desired reliability increases the number of samples must also increase to find at least one infeasible solution. As the number of variables increases, specially for non-linear problems, the MCS approach becomes inadequate for practical use, due to its prohibitively high computation cost.

Since Monte Carlo method is basically a sampling process, the results are subjected to sampling error that decreases with the sample size. However, using procedures known as variance reduction techniques the error may be reduced without increasing the sample size. One of such procedures with a high convergence rate is the Monte Carlo with Importance Sampling (MCIS) [87]. In MCIS, the regions of interest for the simulation process are those around the points in the failure domain having the largest values, i.e., the design points.

Given the simplifications adopted and their approximate nature, none of the above methods provides guarantees on the reliability values computed, specially for non-linear problems. In contrast, the Probabilistic Continuous Constraint framework does not suffer from this limitation, guaranteeing safe bounds for the probability of failure.

## 7.3 Probabilistic Constraint Approach

Reliability analysis can be used to analyze existing systems, thus constituting a rational tool for those in charge of decision-making. On the other hand, reliability calculations can be used in the design process. In this case, it is usually advantageous to combine reliability analysis with optimization algorithms, in order to achieve an optimal and reliable design in view of uncertainties. In the following we describe how the PC framework can be used to obtain safe results on such problem formulations.

### 7.3.1 Reliability Assessment

For the formulation of a reliability assessment problem as a PC, we distinguish between series and parallel systems. The formulation of a series system as a PC follows.

#### **Definition 7.1 (Series System as a Probabilistic Constraint Space)**

*Consider a series system with an associated random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  with joint PDF  $f_{\mathbf{X}}$  defined in  $\Omega_{\mathbf{X}} \subseteq \mathbb{R}^n$  and a set of  $k$  limit-state functions  $g_i$  that define the success event  $S^S$  as in (7.4). The system is modeled as a PC,  $\langle \langle X, D, C \rangle, f \rangle$ , such that:*

$$\begin{aligned} D &\subseteq \Omega_{\mathbf{X}} & X &= \langle x_1, \dots, x_n \rangle \\ C &= \{g_i(\mathbf{x}) \geq 0 : 1 \leq i \leq k\} & f &= f_{\mathbf{X}}(\mathbf{x}) \end{aligned}$$

Its reliability is given by  $P_s^S = P(\mathcal{F}(\langle X, D, C \rangle))$  whereas its probability of failure is  $1 - P_s^S$ .

Likewise, the formulation of a parallel system as a PC is described next.

**Definition 7.2 (Parallel System as a Probabilistic Constraint Space)**

Consider a parallel system with an associated random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  with joint PDF  $f_{\mathbf{X}}$  defined in  $\Omega_{\mathbf{X}} \subseteq \mathbb{R}^n$  and a set of  $k$  limit-state functions that define the failure event  $F^P$  as in (7.5). This system is modeled as a PC,  $\langle \langle X, D, C \rangle, f \rangle$ , such that:

$$\begin{aligned} D &\subseteq \Omega_{\mathbf{X}} & X &= \langle x_1, \dots, x_n \rangle \\ C &= \{g_i(\mathbf{x}) \leq 0 : 1 \leq i \leq k\} & f &= f_{\mathbf{X}}(\mathbf{x}) \end{aligned}$$

Its probability of failure is given by  $P_f^P = P(\mathcal{F}(\langle X, D, C \rangle))$  whereas its reliability is  $1 - P_f^P$ .

From the previous definitions algorithm 6 (page 97) can easily be used to compute the probability of event  $\mathcal{H} = \mathcal{F}(\langle X, D, C \rangle)$  and obtain enclosures for the reliability (or probability of failure) of series or parallel systems. Its arguments are  $\langle \langle X, D, C \rangle, f \rangle$  and  $C_{\mathcal{H}} = C$ .

Notice that reliability problems do not impose bounds on the random variables, which is not possible to model in the PC framework, where  $D \subseteq \Omega_{\mathbf{X}}$  must be a bounded box. Thus, to guarantee the safety of the computed probability enclosures when using the validated quadrature method, a small correction factor must be added to such enclosure. When  $D \subset \Omega_{\mathbf{X}}$  it is necessary to quantify the probability neglected by algorithm 6. First an enclosure for the probability of event  $\mathcal{D} = \mathcal{F}(\langle X, D, \{\} \rangle)$  is computed,  $[P](\mathcal{D})$ . The enclosure for the neglected probability is  $[P](\Omega_{\mathbf{X}} \setminus \mathcal{D}) = 1 - [P](\mathcal{D})$ . Then the term  $[0, \sup([P](\Omega_{\mathbf{X}} \setminus \mathcal{D}))]$  is added to the enclosure computed by algorithm 6.

## Experimental Results

To illustrate the limitations of the classical techniques (FORM, SORM and Monte Carlo) described in section 7.2, several examples of reliability problems found in the

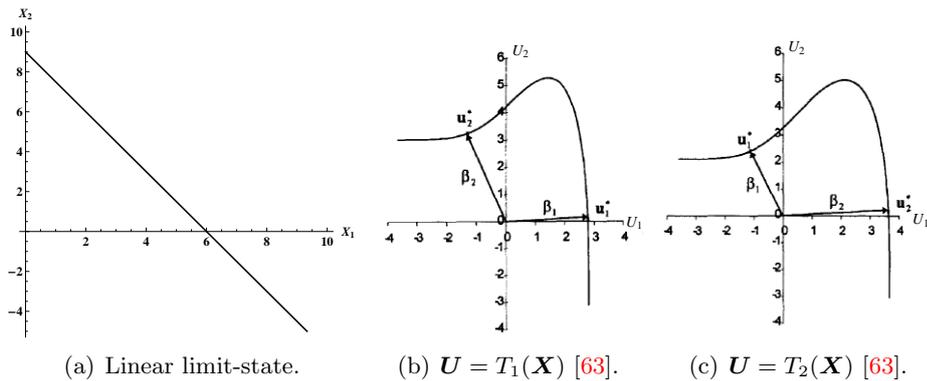
literature are modeled as PCs.

The results obtained with the classical approaches are compared with those computed with the safe version of algorithm 6<sup>1</sup> with  $\delta = 10^{-6}$ ,  $\varepsilon = 10^{-15}$  and a Taylor order of 2, hereafter simply referred as PCTM algorithm. In the experiments, function *NProbability*<sup>2</sup> with the default parametrization, is also used to compute the required probabilities as a complementary source of comparison. All the experiments were performed on an Intel Core Duo at 1.83 GHz with 1 GB of RAM.

The first example illustrates the non linearity induced in the limit-state function resulting from the transformation of a non Gaussian distribution into a standard normal distribution.

**Example 7.1.** Consider the reliability problem, originally introduced in [63], with  $\mathbf{X} = \langle X_1, X_2 \rangle$ , joint PDF  $f_{\mathbf{X}}(x_1, x_2) = (x_1 + x_2 + x_1x_2)e^{-(x_1+x_2+x_1x_2)}$  defined in  $\Omega = [0, \infty[ \times [0, \infty[$  and limit-state function  $g(x_1, x_2) = 18 - 3x_1 - 2x_2$ .

Although the limit-state function is linear in the original space, it becomes highly non linear and has two design points in the standard normal space, due to the strong non normality of the random variables. Figure 7.4 shows the limit-state function (a) in the original space and (b) and (c) in the standard normal space with the transformations, respectively  $T_1(\mathbf{X})$  and  $T_2(\mathbf{X})$ , described in [63].



**Figure 7.4:** Example of a linear limit-state in the original space and non-linear in the standard normal space.

<sup>1</sup>Safe version is a shortcut for the version of the algorithm that uses the verified quadrature method, based on Taylor models, to compute the probability.

<sup>2</sup>From *Mathematica* v8.0.1.0 [119].

The problem is formulated as a PC:

$$\begin{aligned} X &= \langle x_1, x_2 \rangle & D &= [0, 15] \times [0, 15] \\ C &= \{g(x_1, x_2) \leq 0\} & f &= f_{\mathbf{X}}(x_1, x_2) \end{aligned} \quad (7.8)$$

The bounds chosen for  $D$  guarantee a negligible probability for the neglected  $\Omega$  region,  $[P](\Omega \setminus \mathcal{D}) \leq 6.2 \times 10^{-7}$ .

The results obtained with the classical approaches (from [73]) and with the PCTM algorithm (after adding the correction term) are shown in table 7.1. It presents the approximations obtained by FORM and SORM methods when only one of the design points is considered and when both are considered for both transformations  $T_1(\mathbf{X})$  and  $T_2(\mathbf{X})$ . For Monte Carlo (MC) and PCTM algorithm this does not apply.

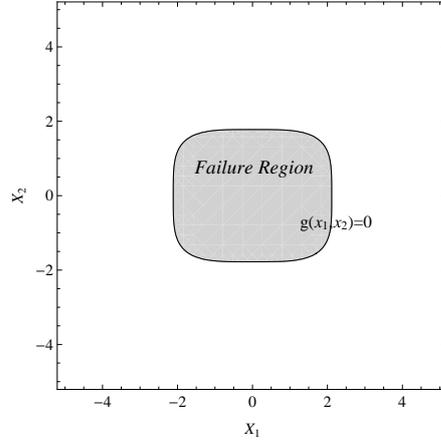
	$u_1^*$ alone		$u_2^*$ alone		$u_1^*$ and $u_2^*$		MC	PCTM
	FORM	SORM	FORM	SORM	FORM	SORM		
$T_1(\mathbf{X})$	0.269	0.279	0.023	0.016	0.292	0.296		
$T_2(\mathbf{X})$	0.404	0.294	0.014	0.015	0.417	0.308	0.294	[0.2943, 0.2946]

**Table 7.1:** Probability of failure  $\times 10^2$ .

It is clear from table 7.1 that the results obtained with FORM and SORM have a great variability, depending on the chosen configuration, and, except for one, are outside the safe enclosure computed by PCTM algorithm (obtained in about 4 seconds CPU time). Using *Mathematica* the obtained result (in less than 1 second CPU time) was  $0.2944 \times 10^{-2}$ , which is in accordance with the enclosure computed by the PCTM algorithm.

The next example illustrates a non linear limit-state function where the original space is normal (although not standard normal).

**Example 7.2.** Consider the reliability problem from [33], with  $\mathbf{X} = \langle X_1, X_2 \rangle$ , where  $X_1 \sim \mathcal{N}(10, 5)$  and  $X_2 \sim \mathcal{N}(10, 5)$  are independent random variables defined in  $\Omega = \mathbb{R}^2$ , and limit-state function  $g(x_1, x_2) = x_1^4 + 2x_2^4 - 20$ . Figure 7.5 shows the limit-state function and the failure region of this problem.



**Figure 7.5:** Non linear limit-state function  $g(x_1, x_2) = 0$ .

The problem is formulated as a PC:

$$\begin{aligned}
 X &= \langle x_1, x_2 \rangle & D &= [-40, 60] \times [-40, 60] \\
 C &= \{g(x_1, x_2) \leq 0\} & f &= \frac{1}{50\pi} e^{-\frac{1}{2} \left[ \left( \frac{x_1-10}{5} \right)^2 + \left( \frac{x_2-10}{5} \right)^2 \right]} \quad (7.9)
 \end{aligned}$$

Since 10 standard deviations around the mean value are assumed, the bounds chosen for  $D$  guarantee a negligible probability for the neglected  $\Omega$  region,  $[P](\Omega \setminus \mathcal{D}) \leq 2.8 \times 10^{-13}$ .

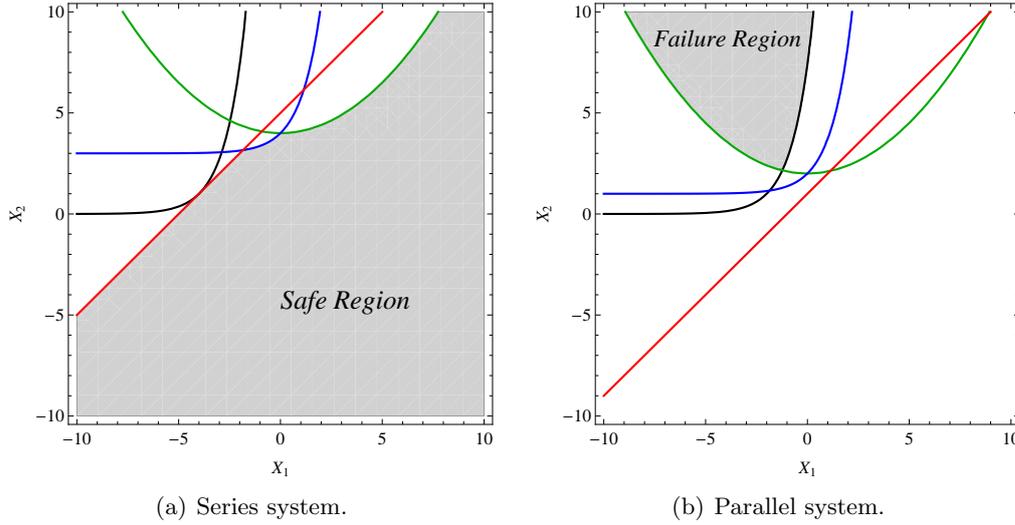
The results obtained with the classical approaches (from [33, pag. 132-136]) where two SORM versions are considered (see [33, Chapter 4] for details) and with the PCTM algorithm (after adding the correction term) are shown in table 7.2.

FORM	SORM Breitung	SORM Tvedt	MC	PCTM
0.9005	0.2221	0.2087	0.1950	[0.1851, 0.1853]

**Table 7.2:** Probability of failure  $\times 10^2$ .

The result obtained with FORM grossly overestimates the probability of failure. Those obtained with both versions of SORM are closer to the correct value, however are still far from the exact value. Simulation with Monte Carlo produces the result closer to the correct value with an 5.23% error. Using *Mathematica* the obtained result (in about 7 seconds CPU time) was  $0.1853 \times 10^{-2}$ , which is in accordance with the enclosure computed by the PCTM algorithm (in about 4 seconds CPU time).

The following examples are found in [111] to illustrate series and parallel systems.



**Figure 7.6:** Examples of series and parallel systems found in [111].

**Example 7.3.** Consider the reliability problem from [111, Note 6], with  $\mathbf{X} = \langle X_1, X_2 \rangle$ , where  $X_1$  and  $X_2$  are independent standard normal random variables defined in  $\Omega = \mathbb{R}^2$ , and the series system defined by the limit state functions:

$$\begin{aligned}
 g_1(x_1, x_2) &= e^{x_1} - x_2 + 3 & g_2(x_1, x_2) &= x_1 - x_2 + 5 \\
 g_3(x_1, x_2) &= e^{x_1+4} - x_2 & g_4(x_1, x_2) &= 0.1x_1^2 - x_2 + 4
 \end{aligned}$$

Figure 7.6 (a) shows the limit-state functions and the safe region of this problem.

The problem is formulated as a PC:

$$\begin{aligned}
 D &= [-10, 10] \times [-10, 10] & X &= \langle x_1, x_2 \rangle \\
 C &= \{g_i(\mathbf{x}, \mathbf{y}) \geq 0 : 1 \leq i \leq 4\} & f &= \frac{1}{2\pi} e^{-\frac{1}{2}(x_1^2+x_2^2)}
 \end{aligned}$$

Again 10 standard deviations around the mean value are assumed for the bounds of  $D$ , with  $[P](\Omega \setminus \mathcal{D}) \leq 2.8 \times 10^{-13}$ .

The results obtained with the classical approach for series systems analysis, where simple (SB) and Ditlevsen (DB) bounds are considered (see [111, Note 6] for details)

and with the PCTM algorithm<sup>1</sup> (after adding the correction term) are shown in table 7.3.

SB	DB	PCTM
[0.2241, 0.5190]	[0.3516, 0.4091]	[0.3124, 0.3135]

**Table 7.3:** Probability of failure  $\times 10^3$ .

The simple bounds are too wide to be informative, while the more accurate Ditlevsen bounds do not include the exact value in the safe enclosure computed by the PCTM algorithm (in about 75 seconds CPU time). The result obtained with *Mathematica* (in about 2 seconds CPU time),  $0.0312989 \times 10^{-2}$ , is in accordance with the enclosure obtained with the PCTM algorithm.

**Example 7.4.** Consider the reliability problem from [111, Note 7], with  $\mathbf{X} = \langle X_1, X_2 \rangle$ , where  $X_1$  and  $X_2$  are independent standard normal random variables defined in  $\Omega = \mathbb{R}^2$ , and the parallel system defined by the limit state functions:

$$\begin{aligned} g_1(x_1, x_2) &= e^{x_1} - x_2 + 1 & g_2(x_1, x_2) &= x_1 - x_2 + 1 \\ g_3(x_1, x_2) &= e^{x_1+2} - x_2 & g_4(x_1, x_2) &= 0.1x_1^2 - x_2 + 2 \end{aligned}$$

Figure 7.6 (b) shows the limit-state functions and the failure region of this problem.

The problem is formulated as a PC:

$$\begin{aligned} D &= [-10, 10] \times [-10, 10] & X &= \langle x_1, x_2 \rangle \\ C &= \{g_i(\mathbf{x}, \mathbf{y}) \leq 0 : 1 \leq i \leq 4\} & f &= \frac{1}{2\pi} e^{-\frac{1}{2}(x_1^2+x_2^2)} \end{aligned}$$

Again 10 standard deviations around the mean value are assumed for the bounds of  $D$ , with  $[P](\Omega \setminus \mathcal{D}) \leq 2.8 \times 10^{-13}$ .

The results obtained with the classical approach for parallel systems analysis, where simple and Ditlevsen bounds are considered (see [111, Note 7] for details) and with the PCTM algorithm (after adding the correction term) are shown in table 7.4. No results were available for the Monte Carlo method.

<sup>1</sup>In fact, the probability of failure was obtained from the reliability computed by the PCTM algorithm.

SB	DB	PCTM
[0.0762, 2.1692]	[0.1264, 0.2256]	[0.1703, 0.1705]

**Table 7.4:** Probability of failure  $\times 10^2$ .

Like in the series system example, the simple bounds are too wide to be informative. In this case, the more accurate Ditlevsen bounds include the safe enclosure computed by the PCTM algorithm (in about 5 seconds CPU time) but are much wider. The result obtained with *Mathematica* (after about 5 seconds CPU time),  $1.7039 \times 10^{-3}$ , is in accordance with the enclosure obtained with the PCTM algorithm.

---

### 7.3.2 Reliability Based Design

When dealing with reliability based design problems, both uncontrollable variables  $\mathbf{X}$  and design parameters  $\mathbf{Y}$  are considered. The reliability of a given design is the probability of success of the modeled system given the choices made on the design parameters, and can be obtained by reliability assessment techniques.

In the context of the PC framework we consider continuous design parameters and a given design  $\Delta$  is an interval instantiation of such parameters (a box), where the values in each interval are indifferent among each other, i.e., are equally likely. Consider, for instance, that a design parameter describes the length of an element in centimeters and adequate values range between 1 and 10. If it is impossible to produce elements with an accuracy of more than one millimeter, then possible instantiations would consider intervals with a granularity of 0.1.

The space  $\Omega_{\mathbf{X}}$  associated with random vector  $\mathbf{X}$  is extended with the design space  $\Omega_{\mathbf{Y}}$ , where the random vector  $\mathbf{Y}$  is assumed uniformly distributed with PDF  $f_{\mathbf{Y}} = \frac{1}{\text{vol}(\Omega_{\mathbf{Y}})}$ . Since design parameters and uncontrollable variables are probabilistically independent the joint PDF  $f : \Omega_{\mathbf{X}} \times \Omega_{\mathbf{Y}} \rightarrow [0, +\infty]$  is (see definition 3.27):

$$f(\mathbf{x}, \mathbf{y}) = f_{\mathbf{X}}(\mathbf{x}) \times f_{\mathbf{Y}}(\mathbf{y})$$

For a single limit-state function, the failure and success events associated with a design  $\Delta \subseteq \Omega_{\mathbf{Y}}$  are<sup>1</sup>:

$$\begin{aligned} F(\Delta) &= \{\langle \mathbf{x}, \mathbf{y} \rangle \in \Omega_{\mathbf{X}} \times \Delta : g(\mathbf{x}, \mathbf{y}) < 0\} \\ S(\Delta) &= \{\langle \mathbf{x}, \mathbf{y} \rangle \in \Omega_{\mathbf{X}} \times \Delta : g(\mathbf{x}, \mathbf{y}) \geq 0\} \end{aligned}$$

The probability of failure (reliability) of design  $\Delta$  is the conditional probability of the failure event (success event) given  $\Delta$ :

$$\begin{aligned} P_f(\Delta) &= P(\langle \mathbf{x}, \mathbf{y} \rangle \in F(\Delta) | \mathbf{y} \in \Delta) = \alpha \int_{F(\Delta)} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} d\mathbf{y} \\ P_s(\Delta) &= P(\langle \mathbf{x}, \mathbf{y} \rangle \in S(\Delta) | \mathbf{y} \in \Delta) = \alpha \int_{S(\Delta)} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} d\mathbf{y} \end{aligned}$$

where  $\alpha = \frac{1}{\text{vol}(\Delta)}$ .

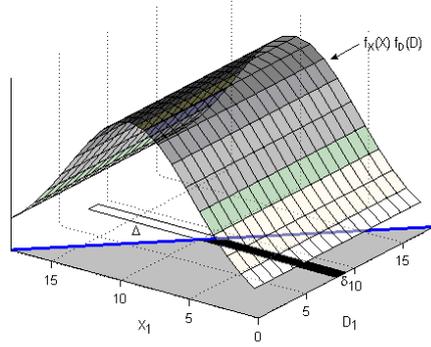
The formula for  $P_f(\Delta)$  can be derived as follows (a similar derivation can be done for  $P_s(\Delta)$ ):

$$\begin{aligned} P_f(\Delta) &= P(\langle \mathbf{x}, \mathbf{y} \rangle \in F(\Delta) | \mathbf{y} \in \Delta) = \frac{P(\langle \mathbf{x}, \mathbf{y} \rangle \in F(\Delta))}{P(\mathbf{y} \in \Delta)} \\ &= \frac{\int_{F(\Delta)} f(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}}{\int_{\Delta} f_{\mathbf{Y}}(\mathbf{y}) d\mathbf{y}} = \frac{\frac{1}{\text{vol}(\Omega_{\mathbf{X}})} \int_{F(\Delta)} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} d\mathbf{y}}{\frac{1}{\text{vol}(\Omega_{\mathbf{Y}})} \int_{\Delta} d\mathbf{y}} \\ &= \frac{\int_{F(\Delta)} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} d\mathbf{y}}{\text{vol}(\Delta)} \end{aligned}$$

**Example 7.5.** Consider the design problem, illustrated in figure 7.7, with an uncontrollable variable  $X_1$  defined in  $\Omega_X = [0, 18]$  with a truncated Gaussian PDF  $f_X$ , a design parameter  $Y_1$  defined in  $\Omega_Y = [0, 18]$  with uniform PDF  $f_Y = \frac{1}{18}$  and a limit-state function  $g(x_1, y_1) = 18 - y_1 - x_1$ . The success event  $S$  (grey area) is defined by  $g(x_1, y_1) \geq 0$ .

One possible decision is  $\Delta = [8, 9]$  (with granularity 1), where  $F(\Delta)$  and  $S(\Delta)$  are, respectively, the white and black areas inside the rectangle. Its probability of failure

<sup>1</sup>When series or parallel systems are considered these events can be adapted based on, respectively, (7.4) and (7.5).



**Figure 7.7:** A design problem.

and reliability are:

$$P_f([8, 9]) = \alpha \int_8^9 \int_{18-y_1}^{18} f_X(x_1) dx_1 dy_1$$

$$P_s([8, 9]) = \alpha \int_8^9 \int_0^{18-y_1} f_X(x_1) dx_1 dy_1$$

where  $\alpha = \frac{1}{\text{vol}([8,9])} = 1$ .

---

Reliability based design of series and parallel systems, can be formulated as probabilistic continuous constraints spaces as follows.

**Definition 7.3 (RBD of a Series System as a PC)** Consider a RBD of a series system with an associated random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  with joint PDF  $f_{\mathbf{X}}$  defined in  $\Omega_{\mathbf{X}} \subseteq \mathbb{R}^n$ , a set of  $m$  design parameters  $Y_i$  assuming values in  $\Omega_{\mathbf{Y}} \in \mathbb{I}\mathbb{R}^m$  and a set of  $k$  limit-state functions  $g_i$  that define the success event of a design  $\Delta \subseteq \Omega_{\mathbf{Y}}$  as:

$$\bigcap_{i=1}^k \{ \langle \mathbf{x}, \mathbf{y} \rangle \in \Omega_{\mathbf{X}} \times \Delta : g_i(\mathbf{x}, \mathbf{y}) \geq 0 \}$$

This RBD of a series system is modeled as a PC,  $\langle\langle X, D, C \rangle, f\rangle$ , such that:

$$\begin{aligned} D &= D_{\mathbf{X}} \times \Omega_{\mathbf{Y}} \text{ where } D_{\mathbf{X}} \subseteq \Omega_{\mathbf{X}} & X &= \langle x_1, \dots, x_n, y_1, \dots, y_m \rangle \\ C &= \{g_i(\mathbf{x}, \mathbf{y}) \geq 0 : 1 \leq i \leq k\} & f &= \frac{f_{\mathbf{X}}(\mathbf{x})}{\text{vol}(\Omega_{\mathbf{Y}})} \end{aligned}$$

The reliability of design  $\Delta$  is given by  $P_s^S(\Delta) = P(\mathcal{F}(\langle X, D_{\mathbf{X}} \times \Delta, C \rangle) | D_{\mathbf{X}} \times \Delta)$  whereas its probability of failure is  $1 - P_s^S(\Delta)$ .

Likewise, the formulation of an RBD of a parallel system as a PC is described next.

**Definition 7.4 (RBD of a Parallel System as a PC)** Consider a RBD of a parallel system with an associated random vector  $\mathbf{X} = \langle X_1, \dots, X_n \rangle$  with joint PDF  $f_{\mathbf{X}}$  defined in  $\Omega_{\mathbf{X}} \subseteq \mathbb{R}^n$ , a set of  $m$  design parameters  $Y_i$  assuming values in  $\Omega_{\mathbf{Y}} \in \mathbb{R}^m$  and a set of  $k$  limit-state functions  $g_i$  that define the failure event of a design  $\Delta \subseteq \Omega_{\mathbf{Y}}$  as:

$$\bigcap_{i=1}^k \{ \langle \mathbf{x}, \mathbf{y} \rangle \in \Omega_{\mathbf{X}} \times \Delta : g_i(\mathbf{x}, \mathbf{y}) < 0 \}$$

This RBD of a parallel system is modeled as a PC,  $\langle\langle X, D, C \rangle, f\rangle$ , such that:

$$\begin{aligned} D &= D_{\mathbf{X}} \times \Omega_{\mathbf{Y}} \text{ where } D_{\mathbf{X}} \subseteq \Omega_{\mathbf{X}} & X &= \langle x_1, \dots, x_n, y_1, \dots, y_m \rangle \\ C &= \{g_i(\mathbf{x}, \mathbf{y}) \leq 0 : 1 \leq i \leq k\} & f &= \frac{f_{\mathbf{X}}(\mathbf{x})}{\text{vol}(\Omega_{\mathbf{Y}})} \end{aligned}$$

The probability of failure of design  $\Delta$  is  $P_f^P(\Delta) = P(\mathcal{F}(\langle X, D_{\mathbf{X}} \times \Delta, C \rangle) | D_{\mathbf{X}} \times \Delta)$  whereas its reliability is  $1 - P_f^P(\Delta)$ .

To obtain a PC that models a single design  $\Delta \subseteq \Omega_{\mathbf{Y}}$ , we simply replace  $\Omega_{\mathbf{Y}}$  by  $\Delta$  in the above definitions and algorithm 6 (page 97) can be used to compute an enclosure for its reliability (or probability of failure), with arguments  $\langle\langle X, D_{\mathbf{X}} \times \Delta, C \rangle, f\rangle$  and  $C_{\mathcal{H}} = C$ .

Similarly to section 7.3.1, when  $\Omega_{\mathbf{X}}$  is unbounded then a sufficiently large bounded box  $D_{\mathbf{X}}$  is considered and a correction term is added to the probability enclosure computed by algorithm 6.

Given the design space  $\Omega_{\mathbf{Y}}$ , the PC framework is able to characterize the reliability of all meaningful designs  $\Delta_i$  within such space, given a granularity specified by the decision maker. Algorithm 12 (page 134) with arguments  $\langle\langle X, D_{\mathbf{X}} \times \Omega_{\mathbf{Y}}, C \rangle, f\rangle$  and  $C_{\mathcal{H}} = C$ , and some slight adaptations, can be used for this purpose.

Algorithm 14 incorporates the necessary modifications to algorithm 12 and furthermore allows an extra parameter that guarantees that only designs with a reliability greater than a given threshold are considered. It computes enclosures for the marginal reliability (or probability of failure) of every meaningful design in  $\Omega_{\mathbf{Y}}$  for a subset of design parameters  $\mathbf{Z} = \langle Y_{i_1}, \dots, Y_{i_m} \rangle$ . The granularity for each considered design parameter is indicated in the corresponding grid spacing  $\alpha_j$ . It outputs an  $m$ -dimensional array  $M$  of reliability enclosures and a box  $H$  that encloses the design space, defining the region characterized by the distribution.

In the pseudo code  $[R]$  represents the computed reliability enclosure for a design which depends on the kind of system addressed. For series systems  $[R](\mathcal{H}_{\boxplus_B}, f) = [P](\mathcal{H}_{\boxplus_B}, f)$  whereas for parallel systems  $[R](\mathcal{H}_{\boxplus_B}, f) = 1 - [P](\mathcal{H}_{\boxplus_B}, f)$ . This is extensive to the probability enclosure used in the  $stop_{\delta \leq}$  criterion.

The algorithm starts by computing the hull of the considered design space, using the grid spacings  $\alpha_j$  and the ranges of the relevant design parameters (lines 1 – 2). This is necessary because such hull may exceed the initial ranges of the design parameters. That excess, not greater than  $\alpha_j$  in each parameters bound, must be accounted for to guarantee the correctness of the computed reliability for designs in the hull boundaries.

The PDF  $f$  given as argument must be modified to compute the reliability of each design. This is not directly included in the original PDF because it depends on the granularity,  $\alpha_i$ , chosen for each design parameter (line 3).

The grid boxes are computed using *cReasoning* (algorithm 3, page 40) with a grid oriented parametrization (see section A.1 in appendix A) (line 4). Then, each grid box (a design) is further refined by another call to *cReasoning* until the required accuracy for its reliability enclosure is achieved or its reliability is assuredly smaller than the given threshold  $tol$  (imposed by  $stop_{\delta \geq}$ ) or every box is already sufficiently small (imposed by  $eligible_{\varepsilon}$ ) (line 7). The reliability of non neglected designs is computed and stored in the corresponding array cell (lines 8 – 12).

**Algorithm 14:** *designDistribution*( $\langle\langle X, D, C \rangle, f\rangle, C_{\mathcal{H}}, Z_{idx}, \varepsilon, \delta, \alpha, tol$ )

**Input:**  $\langle\langle X, D, C \rangle, f\rangle$ : PC;  $C_{\mathcal{H}}$ : set of constraints;  $Z_{idx} = \langle i_1, \dots, i_m \rangle$ : tuple of variables indexes;  $\varepsilon, \delta$ : double;  $\alpha = \langle \alpha_1, \dots, \alpha_m \rangle$ : tuple of doubles;  $tol$ : double

**Output:**  $\langle M, H \rangle$ : pair with an  $m$ -dimensional array of intervals and an  $m$ -box;

- 1  $\forall_{1 \leq j \leq m} (I_j \leftarrow [\text{floor}(\text{inf}(D_{i_j})/\alpha_j), \text{ceil}(\text{sup}(D_{i_j})/\alpha_j)]$ ;  $D_{i_j} \leftarrow \alpha_j I_j$ );
- 2  $H \leftarrow \prod_{Z_{idx}}(D)$ ;
- 3  $vol_{\Delta} \leftarrow \prod_{i=1}^m \alpha_i$ ;  $f \leftarrow f \times \frac{vol(H)}{vol_{\Delta}}$ ;
- 4  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle \leftarrow cReasoning(\langle \{D\}, \emptyset \rangle, C_{\mathcal{H}}, \text{split}_{\alpha}, \text{inner}_d, \text{eligible}_{\alpha}, \text{order}_{\downarrow}, \text{false})$ ;
- 5  $\forall_{1 \leq k_1 \leq \text{wid}(I_1)} \dots \forall_{1 \leq k_m \leq \text{wid}(I_m)} M[k_1] \dots [k_m] \leftarrow [0]$ ;
- 6 **foreach** ( $B \in \mathcal{H}_{\square}$ ) **do**
- 7      $\mathcal{H}_{\boxplus B} \leftarrow cReasoning(\langle \{B\}, \emptyset \rangle, C_{\mathcal{H}}, \text{split}_2, \text{inner}_d, \text{eligible}_{\varepsilon}, \text{order}_P, \text{stop}_{\delta \leq})$ ;
- 8     **if** ( $\text{sup}([R](\mathcal{H}_{\boxplus B}, f)) \geq tol$ ) **then**
- 9          $\forall_{1 \leq j \leq m} B_j \leftarrow \prod_{\langle i_j \rangle}(B)$ ;
- 10          $\forall_{1 \leq j \leq m} k_j \leftarrow \text{ceil}(\text{sup}(B_j)/\alpha_j) - \text{inf}(I_j)$ ;
- 11          $M[k_1] \dots [k_m] \leftarrow [R](\mathcal{H}_{\boxplus B}, f)$ ;
- 12     **end**
- 13 **end**
- 14 **return**  $\langle M, H \rangle$ ;

## Experimental Results

The application of the PC framework to RDB, on two mathematical examples representing series systems from reliability analysis literature (e.g. [38]) follows.

Algorithm 14 was implemented over RealPaver 1.0, and the presented results were obtained using the Taylor model integration technique (PCTM version), where a Taylor model of order 2 was used. The experiments were carried out on an Intel Core i7 CPU at 1.6 GHz.

**Example 7.6.** Consider an RBD of a series system with two design parameters,  $Y_1$  and  $Y_2$ , whose values range over  $\Omega_{\mathbf{Y}} = [1, 10] \times [1, 10]$  and two independent random variables,  $X_1 \sim \mathcal{N}(\mu_1 = 0, \sigma_1 = 0.2)$  and  $X_2 \sim \mathcal{N}(\mu_2 = 0, \sigma_2 = 0.2)$  with joint PDF  $f_X$ , that represent variability around the design values. The limit-state functions are:

$$g_1(x_1, x_2, y_1, y_2) = \frac{1}{20}(y_1 + x_1)^2(y_2 + x_2) - 1$$

$$g_2(x_1, x_2, y_1, y_2) = -(y_1 + x_1)^2 - 8(y_2 + x_2) + 75$$

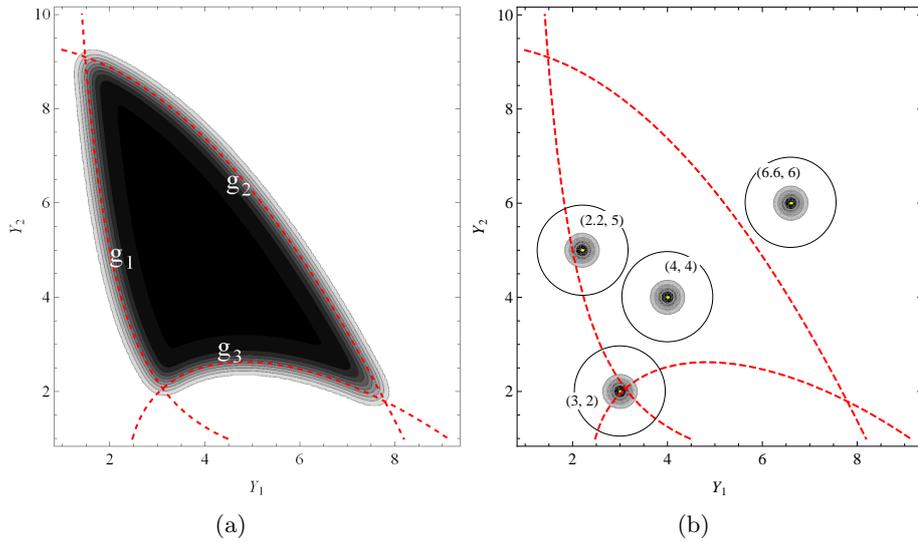
$$g_3(x_1, x_2, y_1, y_2) = 5(y_1 + x_1)^2 + 5(y_2 + x_2)^2 + 6(y_1 + x_1)(y_2 + x_2) - 64(y_1 + x_1) - 16(y_2 + x_2) + 124$$

Such system can be formulated as a PC where:

$$D = [-1, 1] \times [-1, 1] \times \Omega_{\mathbf{Y}} \quad X = \langle x_1, x_2, y_1, y_2 \rangle$$

$$C = \{g_i(\mathbf{x}, \mathbf{y}) \geq 0 : 1 \leq i \leq 3\} \quad f = \frac{f_{\mathbf{X}}(\mathbf{x})}{\text{vol}(\Omega_{\mathbf{Y}})}$$

Since 5 standard deviations around the mean value are assumed, the bounds chosen for  $D_{\mathbf{X}}$  guarantee a negligible probability for the neglected  $\Omega_{\mathbf{X}}$  region,  $[P](\Omega_{\mathbf{X}} \setminus \mathcal{D}_{\mathbf{X}}) \leq 1.2 \times 10^{-6}$ .



**Figure 7.8:** Reliability distribution over the design space.

Algorithm 14 was applied to the above PC, with  $C_{\mathcal{H}} = C$ , with all design parameters ( $Z_{idx} = \langle 3, 4 \rangle$ ), a granularity for each design of  $\alpha_i = 0.05$ , a required precision for the reliability of  $\delta = 5\%$  and a threshold for the reliability of  $tol = 10\%$  (i.e. only designs with a reliability of at least 10% are considered). Figure 7.8 (a) is obtained from  $M$  and  $H$  computed by such call to the algorithm.

The figures for the probability distributions presented in this and other examples are based on the midpoints of the interval enclosures for the grid boxes probabilities returned by algorithm 14 in array  $M$ . In the figures, the more reliable a design is the darker is its color representation (black is 100% reliable whereas white is 0%).

We can observe that reliability is smaller around the boundaries (defined by functions  $g'_i = g_i(\mu_1, \mu_2, y_1, y_2)$ ), and even designs *outside* such boundaries, but close to them, have reliability greater than 0 (although small).

This is because random variables  $X_1$  and  $X_2$  account for some variability of the design variables. Thus a binormal distribution is *centered* on each design (assuming that each design is a point) and it may cover some area within the boundaries representing the safe region.

This idea is illustrated in figure 7.8 (b), where four designs, (2.2, 5), (3, 2), (4, 4) and (6.6, 6), are represented (yellow dots) with circles representing their distribution (regions outside the outer circle have negligible probability). The circle corresponding to design (4, 4) is completely inside the region defined by the boundaries and, as such, has a very high reliability. Design (3, 2) is outside that region but the corresponding circle still has some area inside and so, its reliability is greater than 0. On the other hand design (2.2, 5) is inside the boundary region, but some area of the corresponding circle is outside, decreasing its reliability. Finally, the circle corresponding to design (6.6, 6), completely outside the region, has reliability 0.

Based on figure 7.8 (a) decision makers may choose particular designs of interest and obtain sharp enclosures for their reliability. For a particular design  $(y_1, y_2) = (a, b)$  its reliability can be assessed as explained in section 7.3.1 by considering the PC:

$$\begin{aligned} D &= [-1, 1] \times [-1, 1] & X &= \langle x_1, x_2 \rangle \\ C &= \{g_i(x_1, x_2, a, b) \geq 0 : 1 \leq i \leq 3\} & f &= f_{\mathbf{X}}(\mathbf{x}) \end{aligned}$$

For the decisions illustrated in figure 7.8 (b) the enclosures for their reliability obtained with the PCTM version of algorithm 6 with  $\delta = 10^{-4}$  and  $\varepsilon = 10^{-15}$  (after adding the correction term) are presented on table 7.5. All the results were obtained in less than 10 seconds CPU time.

Design	Reliability
(2.2, 5)	[0.8350, 0.8352]
(3, 2)	[0.1332, 0.1334]
(4, 4)	[0.9999, 1.0000]
(6.6, 6)	[0.0000, 0.0001]

**Table 7.5:** Reliability values for some particular designs.

**Example 7.7.** Consider an RBD of a series system with two design parameters,  $Y_1$  and  $Y_2$ , whose values range over  $\Omega_{\mathbf{Y}} = [-400, 300] \times [-100, 100]$  and two independent random variables,  $X_1 \sim \mathcal{N}(\mu_1 = 0, \sigma_1 = 10)$  and  $X_2 \sim \mathcal{N}(\mu_2 = 0, \sigma_2 = 10)$  with joint PDF  $f_X$ , that represent variability around the design values. The limit-state functions are:

$$\begin{aligned} g_1(x_1, x_2, y_1, y_2) &= (y_1 + x_1)^2 - 1000(y_2 + x_2) \\ g_2(x_1, x_2, y_1, y_2) &= (y_2 + x_2) - (y_1 + x_1) + 200 \\ g_3(x_1, x_2, y_1, y_2) &= (y_1 + x_1) + 3(y_2 + x_2) + 400 \end{aligned}$$

Such system can be formulated as a PC where:

$$\begin{aligned} D &= [-50, 50] \times [-50, 50] \times \Omega_{\mathbf{Y}} & X &= \langle x_1, x_2, y_1, y_2 \rangle \\ C &= \{g_i(\mathbf{x}, \mathbf{y}) \geq 0 : 1 \leq i \leq 3\} & f &= \frac{f_{\mathbf{X}}(\mathbf{x})}{\text{vol}(\Omega_{\mathbf{Y}})} \end{aligned}$$

Again 5 standard deviations around the mean value are assumed for the bounds of  $D_{\mathbf{X}}$ , with  $[P](\Omega_{\mathbf{X}} \setminus D_{\mathbf{X}}) \leq 1.2 \times 10^{-6}$ .

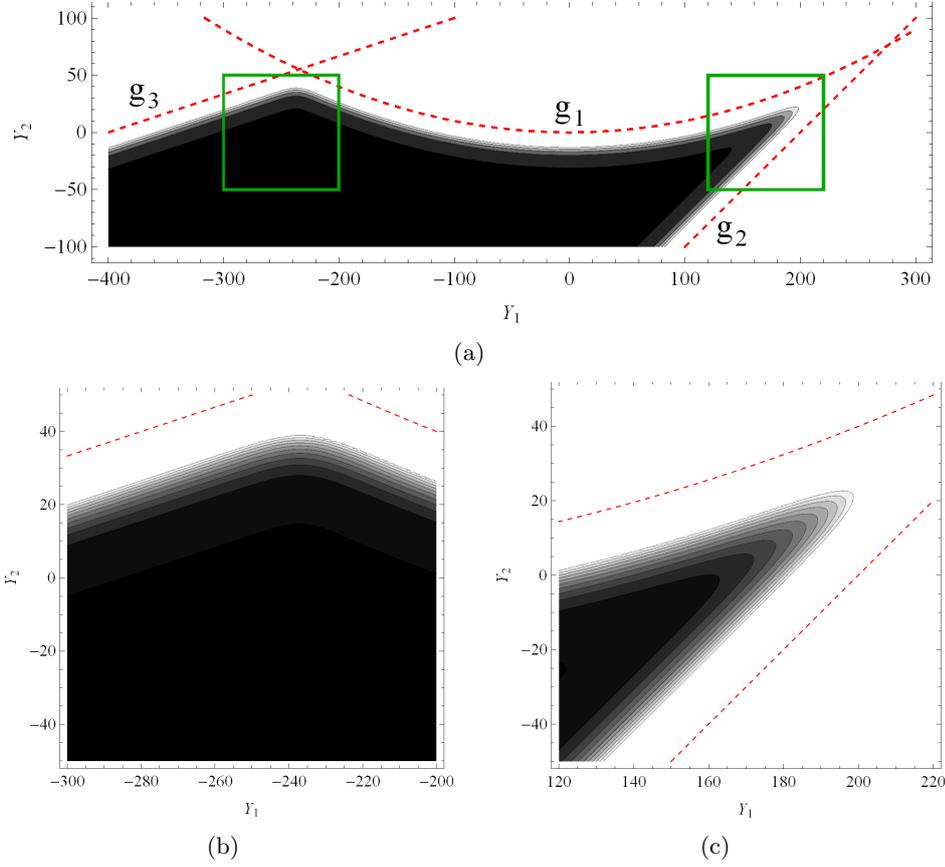
Algorithm 14 is applied to the above PC, with  $C_{\mathcal{H}} = C$ , with all design parameters ( $Z_{idx} = \langle 3, 4 \rangle$ ), a granularity for each design of  $\alpha_i = 1$ , a required precision for the reliability of  $\delta = 5\%$  and a threshold for the reliability of  $tol = 90\%$  (i.e. only designs with a reliability of at least 90% are considered). Figure 7.9 is obtained from the values of  $M$  and  $H$  obtained from such algorithm.

Again, figure 7.9 allows decision makers to have a global view of the problem. Based on this information and on their expertise, they can choose to further explore one or more regions of interest, with increased accuracy.

Figures 7.9 (b) and (c) illustrate a possible *zoom* on two such regions, respectively  $[-300, -200] \times [-50, 50]$  and  $[120, 220] \times [-50, 50]$ , obtained from the results of algorithm 14, with smaller granularity for each design parameter  $\alpha_i = 0.5$  and tighter accuracy requirements  $\delta = 1\%$ .

Such process can be successively applied to guide decision makers on their search for good designs with adequate granularity and confidence on the risk assessment.

---



**Figure 7.9:** Reliability distribution over the design space.

### 7.3.3 Reliability Based Design Optimization

Reliability based design optimization (RBDO) is a particular case of RDB where one or more objective functions, that characterize the desired behavior of the system, are considered. As such, the formulation of an RBDO of a series or a parallel system as a PC is the same as for the RBD of such system. What is inherently different is the information computed in each case.

Our approach to RBDO problems aims at obtaining a Pareto-optimal frontier of non dominated designs wrt the maximization of reliability and the minimization of the expected values of the objective functions. As already stated in section 7.1.1, each objective function  $h_i$  is defined over the uncontrollable variables  $\mathbf{X}$  and the design parameters  $\mathbf{Y}$ , i.e.,  $h_i : \Omega_{\mathbf{X}} \times \Omega_{\mathbf{Y}} \rightarrow \mathbb{R}$ .

Since there is no optimization algorithm available in the PC framework to compute the desired information, algorithm 15, adapted from algorithm 14, is presented here for that purpose.

Besides computing the grid (lines 1 – 4), as in algorithm 14, the new optimization algorithm constructs the set  $L$  of non dominated solutions (lines 5 – 24). Each solution is a pair: the first element is a box (the design); and the second is a vector of values corresponding to the objective functions evaluated at that design. The first position of such vector contains the reliability of the design (line 10). Since minimization is assumed in the Pareto criterion, its complement is considered. The other vector positions contain the conditional expected value of each objective function given the design (line 11).

When a new solution  $\langle B, O \rangle$  is computed, all the designs in  $L$ , dominated by such solution, are excluded from  $L$  and  $\langle B, O \rangle$  is, assuredly, a non dominated solution (lines 13 – 16). Otherwise, if some other design in  $L$  dominates this solution, then it is, assuredly, dominated and it is no longer necessary to further inspect  $L$  (lines 17 – 20). If the new solution  $\langle B, O \rangle$  was not identified as dominated then it is inserted in the set of non dominated solutions  $L$  (line 22).

In lines 25 – 29 the information associated with each non dominated solution in  $L$  is inserted in its corresponding grid cell in array  $M$ .

## Experimental Results

To illustrate the results obtained with algorithm 15<sup>1</sup> the same examples of section 7.3.2 are used, where objective functions are considered.

**Example 7.8.** Consider the PC of example 7.6 and the objective function:

$$h_1(x_1, x_2, y_1, y_2) = y_1 + y_2.$$

Algorithm 15 was applied to this PC, with design parameters ( $Z_{idx} = \langle 3, 4 \rangle$ ),  $h = \langle h_1 \rangle$ , a granularity for each design of  $\alpha_i = 0.1$ , a required precision for the reliability of  $\delta = 5\%$  and a threshold for the reliability of  $tol = 80.134\%$  (two Gaussian standard deviations). The non dominated designs, shown in figure 7.10 (a), were obtained near the feasible region above the intersection of functions  $g'_1$  and  $g'_3$ . Figure 7.10 (b) shows the relation

---

<sup>1</sup>Using the same settings as the experiments in RBD.

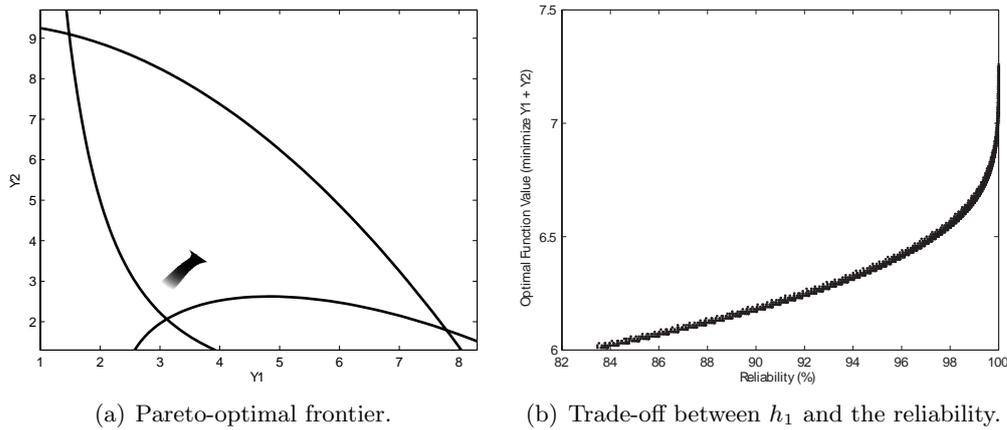
**Algorithm 15:**  $designOptimization(\langle\langle X, D, C \rangle, f \rangle, C_{\mathcal{H}}, Z_{idx}, \varepsilon, \delta, \alpha, tol)$

**Input:**  $\langle\langle X, D, C \rangle, f \rangle$ : PC;  $C_{\mathcal{H}}$ : set of constraints;  $Z_{idx} = \langle i_1, \dots, i_m \rangle$ : tuple of variables indexes;  $h = \langle h_1(X), \dots, h_l(X) \rangle$ : tuple of functions;  $\varepsilon, \delta, tol$ : double;  $\alpha = \langle \alpha_1, \dots, \alpha_m \rangle$ : tuple of doubles;

**Output:**  $\langle M, H \rangle$ : pair with an  $m$ -dimensional array of intervals and an  $m$ -box;

- 1  $\forall_{1 \leq j \leq m} (I_j \leftarrow [\text{floor}(\text{inf}(D_{i_j})/\alpha_j), \text{ceil}(\text{sup}(D_{i_j})/\alpha_j)]$ ;  $D_{i_j} \leftarrow \alpha_j I_j$ );
- 2  $H \leftarrow \prod_{Z_{idx}} (D)$ ;
- 3  $vol_{\Delta} \leftarrow \prod_{i=1}^m \alpha_i$ ;  $f \leftarrow f \times \frac{vol(H)}{vol_{\Delta}}$ ;
- 4  $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle \leftarrow cReasoning(\langle\{D\}, \emptyset \rangle, C_{\mathcal{H}}, split_{\alpha}, inner_d, eligible_{\alpha}, order_{\downarrow}, false)$ ;
- 5  $\forall_{1 \leq k_1 \leq wid(I_1)} \dots \forall_{1 \leq k_m \leq wid(I_m)} M[k_1] \dots [k_m] \leftarrow \emptyset$ ;
- 6  $L \leftarrow \emptyset$ ;
- 7 **foreach** ( $B \in \mathcal{H}_{\square}$ ) **do**
- 8      $flag \leftarrow unknown$ ;
- 9      $\mathcal{H}_{\boxplus B} \leftarrow cReasoning(\langle\{B\}, \emptyset \rangle, C_{\mathcal{H}}, split_2, inner_d, eligible_{\varepsilon}, order_P, stop_{\delta})$ ;
- 10    **if** ( $\text{sup}([P]')(\mathcal{H}_{\boxplus B}, f) \geq tol$ ) **then**
- 11      $O[0] \leftarrow -1 \times [P]'(\mathcal{H}_{\boxplus B}, f)$ ;
- 12      $\forall_{1 \leq i \leq l} O[i] \leftarrow [E](h_i, f | \mathcal{H}_{\boxplus B})$ ;
- 13     **foreach** ( $\langle B', O' \rangle \in L$ ) **do**
- 14         **if** ( $\text{dominates}(O, O')$ ) **then**
- 15              $L \leftarrow L \setminus \langle B', O' \rangle$ ;
- 16              $flag = nonDominated$ ;
- 17         **end**
- 18         **if** ( $flag = unknown \wedge \text{dominates}(O', O)$ ) **then**
- 19              $flag = dominated$ ;
- 20             **break**;
- 21         **end**
- 22     **end**
- 23     **if** ( $flag \neq dominated$ ) **then**  $L \leftarrow L \cup \{\langle B, O \rangle\}$ ;
- 24    **end**
- 25 **end**
- 26 **foreach** ( $\langle B, O \rangle \in L$ ) **do**
- 27      $\forall_{1 \leq j \leq m} B_j \leftarrow \prod_{\langle i_j \rangle} (B)$ ;
- 28      $\forall_{1 \leq j \leq m} k_j \leftarrow \text{ceil}(\text{sup}(B_j)/\alpha_j) - \text{inf}(I_j)$ ;
- 29      $M[k_1] \dots [k_m] \leftarrow \langle B, O \rangle$ ;
- 30 **end**
- 31 **return**  $\langle M, H \rangle$ ;

between the reliability values and the corresponding  $h_1$  values for the obtained designs. This provides, to decision makers, important information on the trade-off between the system reliability and its optimal behavior.



**Figure 7.10:** Reliability based design optimization.

While the PC framework outputs a safe Pareto-optimal frontier, classical RBDO techniques output a single design point (corresponding to the optimum value of the objective function) given a target reliability, with no guarantees of global optimality.

The PC framework can easily incorporate a method to obtain the best design (or set of designs) given the objective functions, for a target reliability. For instance, for target reliability 84.134% the global optimum for objective function  $h_1$  is proved to be enclosed in  $[6.00, 6.02]$ . It was proved that there are no designs with reliability 84.134% and  $h_1$  values less than 6.00 and a particular design was found with  $Y_1 = 3.11$  and  $Y_2 = 2.91$  with the desired reliability (in fact, proved to be above 84.147%), such that  $h_1(3.11, 2.91) = 6.02$ .

Although the PC framework (with the PCTM version of the algorithm) is considerably slower than the classical techniques it provides more information, being a good trade-off between efficiency and guaranteed safe results.

---

For the same series system consider another objective function.

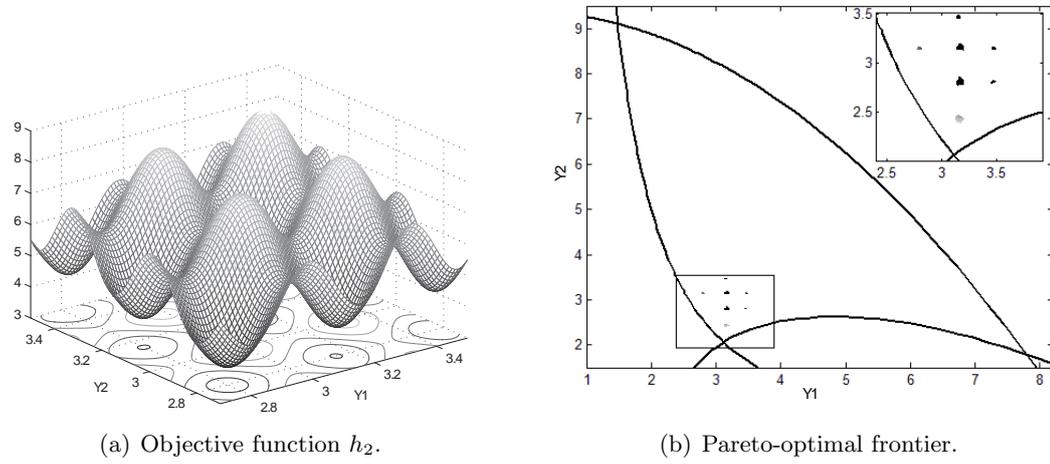
**Example 7.9.** Consider the PC of example 7.6 and the objective function:

$$h_2(x_1, x_2, y_1, y_2) = y_1 + y_2 + \sin(3y_1^2) + \sin(3y_2^2)$$

with several local optima, shown in figure 7.11 (a).

Algorithm 15 was applied to this PC, with design parameters ( $Z_{idx} = \langle 3, 4 \rangle$ ),  $h = \langle h_2 \rangle$ , a granularity for each design of  $\alpha_i = 0.1$ , a required precision for the reliability of  $\delta = 5\%$  and a threshold for the reliability of  $tol = 80.134\%$  (two Gaussian standard deviations).

The results of the algorithm allow to identify and characterize the local optima of objective function  $h_2$  wrt their reliability, producing an overview of the Pareto-optimal frontier, shown in figure 7.11 (b).



**Figure 7.11:** Reliability based design optimization.

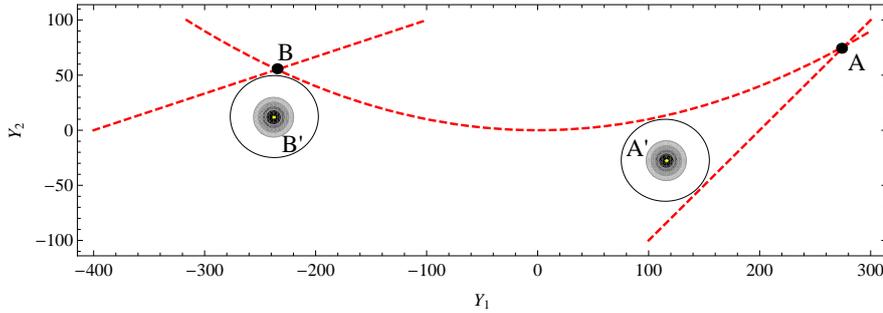
**Example 7.10.** Consider the PC of example 7.7 and the objective function:

$$h_3(x_1, x_2, y_1, y_2) = -y_2$$

Typically, the optimal design lies on a constraint boundary or at the the intersection of more than one constraint, as shown in figure 7.12. When uncertainty is considered in design variables, such optimal design has a very low reliability (see example 7.6). To satisfy a given reliability target, unreliable optimal designs must be sacrificed and designs well within the safe region should be chosen.

As can be observed in the figure, for this problem and objective function  $h_3$  two deterministic optimal designs exist  $A$  (is the global optimum) and  $B$  (is a local optimum).

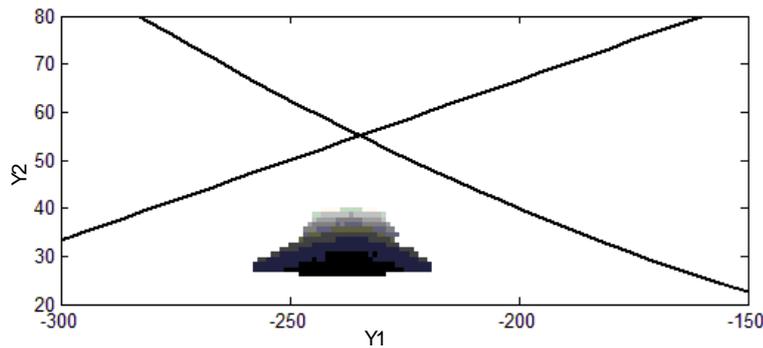
Nevertheless, when considering uncertainty in design variables, design  $B'$  (near the local optimum  $B$ ) is better wrt objective function  $h_3$  and a given target reliability, than design  $A'$  (near the global optimum) for the same reliability.



**Figure 7.12:** Multiple optimal designs.

Algorithm 15 was applied to the PC, with design parameters ( $Z_{idx} = \langle 3, 4 \rangle$ ),  $h = \langle h_3 \rangle$ , a granularity for each design of  $\alpha_i = 0.5$ , a required precision for the reliability of  $\delta = 5\%$  and a threshold for the reliability of  $tol = 90\%$ .

The results of the algorithm allow to identify and characterize the optimum region of objective function  $h_3$  wrt their reliability, producing an overview of the Pareto-optimal frontier, shown in figure 7.13. As expected, the non dominated designs obtained were able to eliminate every design in the region near the deterministic global optimum  $A$ .



**Figure 7.13:** Pareto-optimal frontier

## 7.4 Short Rectangular Column

Consider the engineering problem (a common benchmark used in reliability based design optimization [77]) where the goal is to determine the depth  $h$  and width  $b$  of a short column (the design parameters) with rectangular cross section, where  $15 \leq h \leq 25$  and  $5 \leq b \leq 15$ .

The random variables are the bending moment  $M \sim \mathcal{N}(\mu_1 = 2000, \sigma_1 = 400)$ , the yield stress  $P \sim \mathcal{N}(\mu_2 = 500, \sigma_2 = 100)$  (with correlation  $\rho = 0.5$ ) and the axial force  $Y \sim \text{LogN}(\mu_3 = 5, \sigma_3 = 0.5)$ <sup>1</sup> with joint PDF  $f_A(M, P, Y) = f_{MP}(M, P) \times f_Y(Y)$  where:

$$f_{MP}(M, P) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)} \left[ \left( \frac{M-\mu_1}{\sigma_1} \right)^2 - 2\rho \left( \frac{M-\mu_1}{\sigma_1} \right) \left( \frac{P-\mu_2}{\sigma_2} \right) + \left( \frac{P-\mu_2}{\sigma_2} \right)^2 \right]}$$

$$f_Y(Y) = \frac{1}{Y\sigma_3\sqrt{2\pi}} e^{-\frac{1}{2} \left( \frac{\ln(Y)-\mu_3}{\sigma_3} \right)^2}$$

The limit-state function is:

$$g(M, P, Y, h, b) = 1 - \frac{4M}{(bh^2Y)} - \frac{P^2}{(bhY)^2}$$

The objective is to minimize the total mass given by  $o(h, b) = hb$ . The results reported in [4, 32, 73, 77] refer that, for an allowed probability of failure of 0.00621 (i.e., a target reliability of 0.99379), the optimal design is  $(h, b) = (25, 8.668)$ , with objective function value  $hb = 216.7$ .

This problem can be formulated as PC ColumnA where:

$$D = [0, 4000] \times [0, 1000] \times [10, 2000] \times [15, 25] \times [5, 15] \quad X = \langle M, P, Y, h, b \rangle$$

$$C = \{g(M, P, Y, h, b) \geq 0\} \quad f = \frac{f_A(M, P, Y)}{\text{vol}([15, 25] \times [5, 15])}$$

Algorithm 14 was applied to this PC, with  $C_{\mathcal{H}} = C$ , with design parameters ( $Z_{idx} = \langle 4, 5 \rangle$ ), a granularity for each design of  $\alpha_i = 0.5$ , a required precision for the reliability of  $\delta = 10\%$  and a threshold for the reliability of  $tol = 30\%$ .

<sup>1</sup>LogN represents the lognormal distribution.

Contrary to what was expected, from RBDO references (henceforth denoted as RBDO techniques), all the designs within  $[15, 25] \times [5, 15]$  had a very high reliability.

Using the PC framework for reliability assessment, the reliability of design  $(h, b) = (15, 5)$ , for which  $hb = 75$  is minimum, is above 0.9999 (for the allegedly optimal design  $(h, b) = (25, 8.668)$  it is also above 0.9999). This value clearly satisfies the constraints for the allowed probability of failure and the optimization criterion. Nevertheless, in all reliability analysis studies (to which we had access) this is never the optimal design found.

To try to reproduce the results obtained by the RBDO techniques, a different problem configuration is considered where  $Y \sim \mathcal{N}(\mu_3 = 5, \sigma_3 = 0.5)$  and no correlation exists between random variables  $M$  and  $P$ . The joint PDF is:

$$f_B(M, P, Y) = \frac{1}{2\pi\sqrt{2\pi}\sigma_1\sigma_2\sigma_3} e^{-\frac{1}{2}\left[\left(\frac{M-\mu_1}{\sigma_1}\right)^2 + \left(\frac{P-\mu_2}{\sigma_2}\right)^2 + \left(\frac{Y-\mu_3}{\sigma_3}\right)^2\right]}$$

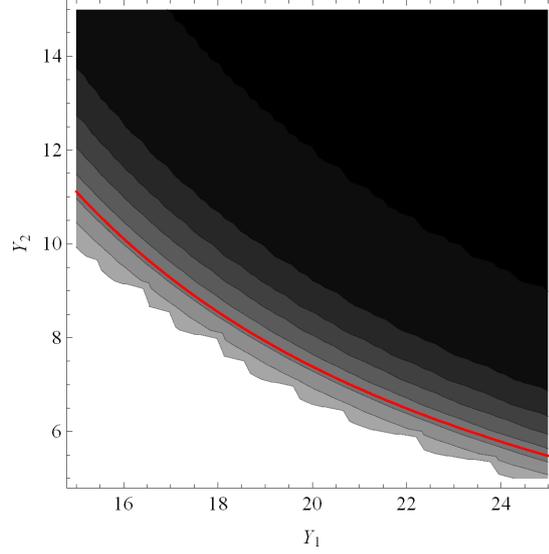
Such problem can be formulated as PC ColumnB where:

$$\begin{aligned} D &= [0, 4000] \times [0, 1000] \times [10, 2000] \times [15, 25] \times [5, 15] & X &= \langle M, P, Y, h, b \rangle \\ C &= \{g(M, P, Y, h, b) \geq 0\} & f &= \frac{f_B(M, P, Y)}{\text{vol}([15, 25] \times [5, 15])} \end{aligned}$$

Using the PC framework for reliability assessment, the reliability of design  $(h, b) = (15, 5)$  is now below 0.0001 and for design  $(h, b) = (25, 8.668)$  it is above 0.9957. These results agree with those obtained by the RBDO techniques.

Figure 7.14 shows the results computed by algorithm 14 applied to PC ColumnB (with the same parameters as above). The dotted red line plots function  $g(\mu_1, \mu_2, \mu_3, h, b)$  where the random variables were replaced by their means (in Gaussian space).

Since in the original configuration the left bounds of each design parameter provided the optimal reliable design, we searched for better designs within smaller bounds. Thus, another problem configuration is considered where the random variables have the same joint PDF,  $f_A$ , but with different bounds for  $h$  and  $b$ :  $3 \leq h \leq 15$  and  $0 \leq b \leq 5$ .



**Figure 7.14:** Reliability distribution over the design space for problem ColumnB.

Such problem can be formulated as PC ColumnC where:

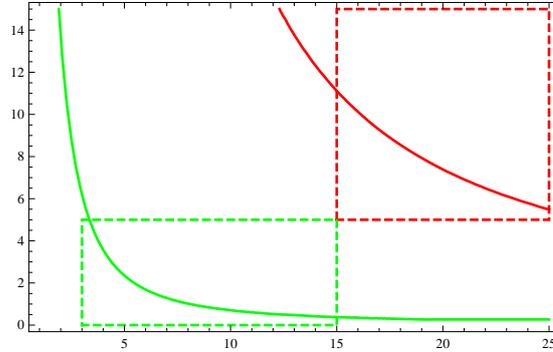
$$D = [0, 4000] \times [0, 1000] \times [10, 2000] \times [3, 15] \times [0, 5] \quad X = \langle M, P, Y, h, b \rangle$$

$$C = \{g(M, P, Y, h, b) \geq 0\} \quad f = \frac{f_A(M, P, Y)}{\text{vol}([3, 15] \times [0, 5])}$$

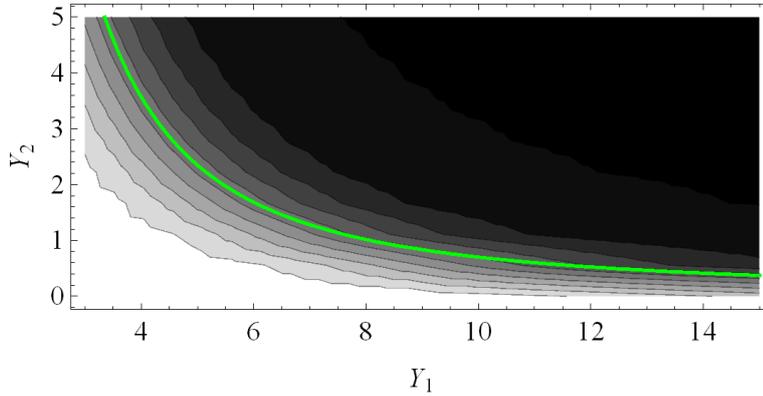
Figure 7.15 plots  $g$  with random variables  $M$ ,  $P$  and  $Y$  replaced by their means (in Gaussian space). In problem ColumnB they are, respectively, 2000, 500 and 5, and the plot of  $g(2000, 500, 5, h, b)$  is the red line. In problem ColumnC they are, respectively, 2000, 500 and  $e^5$ , and the plot of  $g(2000, 500, e^5, h, b)$  is the green line. The dotted squares represent the bounds for the design parameters in problem ColumnB (red) and ColumnC (green).

Algorithm 14 was then applied to PC ColumnC (with the same parameters as above) and the obtained results are presented in figure 7.16.

It can now be observed from the figure, where darker regions represent designs with higher reliability, that not all design have the same high reliability. Using the PC framework for reliability assessment we tested some promising designs (in the black region and with small values for the objective function  $hb$ ) and found design  $(h, b) = (15, 1.7)$  with the objective function value  $hb = 25.5$  and reliability above 0.9978.



**Figure 7.15:** Plots of  $g(2000, 500, 5, h, b)$  (red line) and  $g(2000, 500, e^5, h, b)$  (green line).



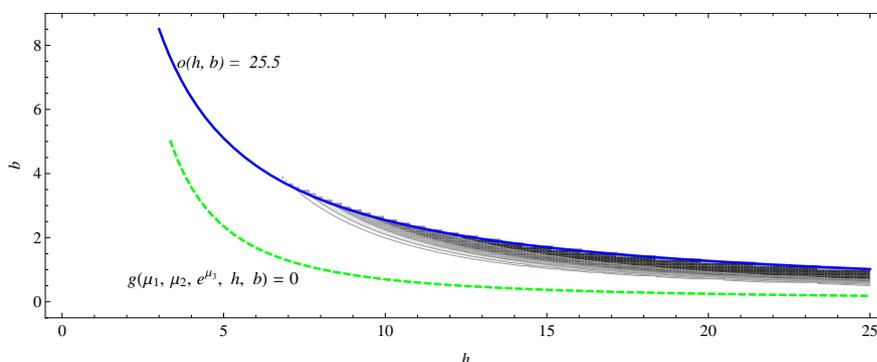
**Figure 7.16:** Reliability distribution over the design space for problem ColumnC.

In order to search for other designs with reliability above 0.99379 and objective function value less or equal than 25.5 a new problem, ColumnD, was formulated:

$$\begin{aligned}
 D &= [0, 4000] \times [0, 1000] \times [10, 2000] \times [0, 25] \times [0, 15] & X &= \langle M, P, Y, h, b \rangle \\
 C &= \{g(M, P, Y, h, b) \geq 0, (h - \alpha_1)(b - \alpha_2) \leq 25.5\} & f &= \frac{f_A(M, P, Y)}{\text{vol}([0, 25] \times [0, 15])}
 \end{aligned}$$

The extra constraint in  $C$  discards boxes where all corresponding objective function values are assuredly above 25.5. The correction terms  $\alpha_i$  correspond to the granularity of each design parameter and is used to avoid narrowing design boxes where not all corresponding objective function values are above 25.5. Such narrowing would bias the reliability computation of those boxes.

Algorithm 14 was applied to PC ColumnD, with  $C_{\mathcal{H}} = C$ ,  $Z_{idx} = \langle 4, 5 \rangle$ ,  $\alpha_i = 0.1$ ,  $\delta = 5\%$  and  $tol = 0.99379$ . The results obtained are presented in figure 7.17. Clearly only a narrow strip of sufficiently reliable designs satisfy the imposed constraints. The region near the upper bound of design parameter  $h$  and the lower bound of design parameter  $b$  is the most promising one. So we considered  $h = 25$  and performed a dichotomic search (using the PC framework for reliability assessment) for  $b$  values between 0.5 and 1.0. An optimal design  $(h, b) = (25, 0.7)$  was found, with objective function value  $hb = 17.5$  and reliability above 0.9939.



**Figure 7.17:** Reliability distribution over the design space for problem ColumnD.

This study shows the suitability of the PC framework for RBDO analysis, which allows directly addressing the original problem formulation with no simplifying assumptions. Moreover, it illustrates the magnitude of the errors that can be made when approximations for probability distributions are considered.

## 7.5 Summary

This chapter illustrated the application of the probabilistic continuous constraint framework to reliability analysis problems. Reliability analysis problems were presented, classical techniques to solve them were addressed and some of their drawbacks discussed. The formulation of reliability analysis problems as probabilistic continuous constraint spaces was presented and the advantages of the PC framework were illustrated on a set of application problems.



## Conclusions and Future Work

The main contribution of this thesis is to extend the classical continuous constraint paradigm with probabilistic reasoning, since this was an important feature for applications in science and engineering. In particular three main tasks were established: the formalization of the probabilistic continuous constraint paradigm to integrate constraint reasoning and probabilistic reasoning; the implementation of a prototype to test the algorithms proposed within the paradigm; and the identification of significant problems in science or engineering that could benefit with the proposed techniques.

### 8.1 Probabilistic Constraint Programming

To formalize the probabilistic constraint paradigm, a generic description and some preliminary results were published in [23], while a more formal characterization was published in [24].

In this thesis the proposed probabilistic continuous constraint paradigm was completely formalized and theoretically grounded by characterizing its affinity with probability theory and with classical methods from constraint programming.

The core of the paradigm, the probabilistic continuous constraint space, was defined and its semantics specified, the queries that can be formulated within such probabilistic space were identified and the concepts of probabilistic constraint event and random vector were introduced. Both safe and approximate methods to obtain enclosures were described (and their properties discussed) namely for (a) conditional and unconditional

probabilities of events; (b) conditional and unconditional expected values and covariance matrices of random vectors; and (c) probability distributions of random vectors.

Besides constraint reasoning, a core component of the framework is the computation of multi dimensional integrals in possibly non linear regions. Two distinct methods were presented to achieve this goal, although other existing techniques could be incorporated.

A safe (or validated) integration method based on Taylor models was theoretically characterized, implemented and tested in the context of the framework. Its convergence analysis is mainly due to Alexandre Goldsztejn and is not a contribution of the present work, being presented here for completeness [48].

The need for a more efficient method that can produce results more rapidly, although with no safety guarantees, motivated the development of an approximate Monte Carlo integration method, hybridized with constraint programming (sampling is only performed in regions not discarded by constraint programming). This was particularly useful in the application of the framework to ocean color inversion. Its success strongly relies on the hybridization, since even a non-naïf Monte Carlo (stratified) method revealed not only to be hard to tune but also impractical in small error settings. This is a promising approach that deserves to be further explored in other application areas.

Since the Monte Carlo integration method appeared at a latter stage of this research, the theoretical characterization of the framework mostly assumes safe integration methods (although the impact of adopting the approximate Monte Carlo method is still addressed) and some of the experimental results do not include Monte Carlo integration (e.g. in the application of the framework to reliability analysis).

The validated version of the framework, for probability assessment and calculation of expected values and variances of random vectors, was tested on a set of mathematical problems, for a proof of concept. Its robustness when dealing with (a) events defined by highly non linear regions, (b) highly non linear PDFs and (c) events with a small probability of occurrence became evident in the set of experiments. The results of probability assessment were compared with those obtained with *Mathematica*, a powerful tool of reference. Clearly, the calculation of probabilities for rare events is very problematic with such tool and may even be considered useless on some of the cases we tested.

Experiments using the approximate hybridized Monte Carlo integration method revealed to be a good trade-off between safety and efficiency, producing, significantly faster, results with only a very small error.

A dedicated algorithm to compute conditional probabilities, exploiting the commonalities of the involved events, was developed and tested on the set of mathematical problems, where results obtained were better than those obtained with a cruder approach that computes separately the probabilities of the conditioning and conditioned events.

Computing the probability of disjunctions of events can also be done within the framework. However, in the future, it would be interesting to provide a more efficient algorithm that accounts for common regions of the events.

The thesis also presents an algorithm to compute nonparametric probability distributions of random vectors and output them in tabular form, for subsequent processing. This is an important feature of the framework, mainly on real world problems such as inverse and reliability problems. In such problems, plots of the results obtained provide an adequate display of the shape of the computed distributions and convey extra information such as the location of the values, its scale, its skewness and whether multiple modes in the distribution exist.

A potential limitation of the framework is its scalability to more complex problems, with higher dimensions, given the continuous constraint paradigm that supports it, as well as the used integration methods. For some classes of problems and for high accuracy requirements, the time necessary to obtain the desired results may drastically increase or even become computationally prohibitive. In the future, experimental research on this subject should be performed, to adequately characterize the behavior of the algorithms in several classes of problems and to seek more efficient algorithms.

Further, attention should be paid in the future, in identifying the influence of different algorithm parameterizations (e.g. accuracy of the computed probability enclosure, minimum authorized width for the boxes, Taylor order or number of samples) in the obtained results.

In the present framework uncertainty is treated by assuming probability distributions in all the variables of the problem. However, in many practical engineering applications,

distributions of some random variables may not be precisely known or uncertainties may not be appropriately represented with distributions and are only known to lie within specified intervals. In the future the probabilistic continuous constraint paradigm can be extended to address problems where uncertain variables are characterized by a mixture of probability distributions and intervals.

Moreover, in some practical engineering applications, both continuous and discrete random variables coexist. The extension of the framework to combine both types of random variables would amplify the range of problems that can be tackled.

## 8.2 Prototype

All the algorithms presented in this thesis were implemented in C++, over the interval-based solver RealPaver 1.0 [52]. This solver provides a set of useful continuous constraint methods and its design makes it easily extensible. The new probabilistic functionalities were incorporated, following a similar design.

An operational prototype application was built, that can readily be used for testing new problems, with minimum configuration efforts. In fact, each algorithm corresponds to an executable program to which a set of adequate arguments is given as input. This process could be further automated, allowing the user to choose the desired functionality within a unique executable.

In general, the results obtained by a given algorithm with an initial parametrization, can guide the user in future parameterizations, allowing some control over the reasoning process (although not during ongoing search): circumscribe reasoning to specific regions of the search space; analyze the robustness of a particular solution; or redefine some input parameters.

In the future, we seek to develop a unique (executable) prototype, requiring minimum configuration efforts, that can be made available to the research community (by disclosing the source code), with proper programmer and user manuals. This is of great importance as it may allow fruitful collaborations with other research groups, result in future extensions to this work and bring to light further application problems.

## 8.3 Application to Decision Problems

Two classes of decision problems were identified as adequate to apply the probabilistic continuous constraint framework: non linear inverse problems and reliability problems. In both cases nonlinear continuous instances exist, with uncertain information characterized by probability distributions.

Three papers on this subject were published: [25] that discusses the application of (a preliminary version of) the framework to solve inverse problems; [26] that explains the application of the framework to reliability problems and how it can be extended to deal with reliability-based optimization problems; and [27] that addresses the capabilities of the framework to deal with decision problems in the presence of uncertainty and non-linearity. A journal paper that addresses the application of the framework to a real world inverse problem, ocean color inversion, was recently submitted [28].

On the two classes of problems, the potentiality of the probabilistic continuous constraint framework was validated in a systematic way. First by identifying the major drawbacks of classical techniques, for each class of problems. Then by testing the framework on typical problematic instances, certifying its robustness.

In the context of inverse problems, ocean color inversion was modeled within the framework. Preliminary results presented to Ocean Color domain experts confirmed the relevance of improving methods to control error propagation in the adopted semi-analytical model, an important issue for decisions about the sensors used in satellite-based studies. The certified results obtained are a first step towards a tighter collaboration.

In reliability based design optimization, a common engineering benchmark was studied (short rectangular column), where the advantages of the framework were clearly highlighted. Its ability to directly address the original problem formulation without simplifying assumptions, avoided errors that are inevitable when approximations for probability distributions are considered.

In the problems addressed, the implemented framework was assessed in terms of accuracy and robustness, not in terms of computation times. Usually, to obtain such guaranteed results and to characterize the complete search space, requires some overhead in the computation time when compared with other methods that only produce

approximate results. Although such overheads could be kept within reasonable bounds, further comparisons should be made to fully assess these differences.

---

This work was motivated by the recognition that constraint solving alone is insufficient to model real life problems, where handling of uncertainty is unavoidable. The pure constraint programming paradigm may handle such uncertainty representing variables within safe domains, but does not distinguish different likelihoods for values within such intervals, leading to decision support poorly informed.

This thesis extends the continuous constraint programming with probability information and is a step forward a better modeling of such problems within constraint programming. The resulting probabilistic constraint framework is thoroughly justified, and its adequacy shown in some realistic engineering problems.

The research now presented opens several opportunities for further research, discussed in this last chapter. We expect to continue research along these lines, so as to more strongly establish the probabilistic constraint framework as a competitive alternative to classical techniques in engineering design problems (and possibly in a wider scope) to combine safety and uncertainty in such problems.

# Appendices



## APPENDIX A

# Constraint Reasoning Algorithm

This appendix presents alternatives for parameterizing algorithms *crStep* (algorithm 2) and *cReasoning* (algorithm 3) as well as the convergence analysis of the last.

## A.1 Parametrization

This section presents alternatives, used in this thesis, for parameterizing algorithms 2 and 3 (in chapter 2, pages 39 and 40), indicating implementations for the *inner*, *eligible* and *stop* predicates, the *split* function and the *order* criterion. It is meant to be used as a reference guide, so the context of each implementation is omitted and explained locally in the part of the thesis that references this appendix.

The arguments of the generic *inner* predicate are a box and a set of constraints and of the *eligible* predicate and *split* function is a box, nevertheless some of their implementations require more arguments. For the sake of simplicity and generality the specification of those other implementations omit the extra arguments, leaving those details for the real implementation.

### Inner Predicate

The *inner* predicate verifies if a box  $B$  is an inner box wrt a set of constraints under specific conditions (e.g., when the box is sufficiently small).

Its default parametrization,  $inner_d$ , relies on natural inclusion functions of the functions induced from the constraints relations, replacing the variables by the intervals of the

box, and checking whether all values in the resulting interval are solutions for the set of constraints. Algorithm 16 presents the respective pseudo code and example A.1 illustrates the concept.

**Algorithm 16:**  $inner_d(B, C)$ 

<p><b>Input:</b> <math>B</math>: box; <math>C</math>: set of constraints;  <b>Output:</b> <math>flag</math>: boolean;</p> <pre> 1 if <math>\exists (s, \rho) \in C</math> <math>isEquation(\rho)</math> then return <math>false</math>; 2 foreach <math>((s, \rho) \in C)</math> do 3   <math>f \leftarrow getFunction(\rho)</math>; 4   <math>I \leftarrow [f]_N(B)</math>; 5   if <math>(\bar{I} &gt; 0)</math> then return <math>false</math>; 6 end 7 return <math>true</math>;                 </pre>
--

In the algorithm function  $isEquation$  verifies if the constraint relation is an equation, function  $getFunction$  retrieves the mathematical function  $f$  induced from the constraint relation  $\rho$  and  $[f]_N$  represents its natural inclusion function. When at least one constraint is an equation there are no inner boxes and the algorithm returns  $false$ . When all constraints are inequalities, if the upper bound of  $[f]_N(B)$  is greater than 0<sup>1</sup> for at least one constraint,  $B$  is not an inner box. Otherwise  $B$  is an inner box.

**Example A.1.** Consider the constraint  $C_1$  whose relation  $\rho$  is given by  $x + y - 5 \leq 0$  and the box  $B = [1, 2] \times [1, 2]$ . Function  $inner_d(B, \{C_1\})$  verifies if  $[1, 2] + [1, 2] - [5] \leq 0 \Leftrightarrow [-3, -1] \leq 0$ . Since every value in the interval  $[-3, -1]$  is less or equal to 0,  $B$  is an inner box.

---

In section 5.4 a grid oriented implementation for the *inner* predicate is required,  $inner_\alpha$ , such that  $inner_\alpha(B, C) \equiv inner_d(B, C) \wedge insideGrid([g_1](B) \times \dots \times [g_m](B), \langle \alpha_1, \dots, \alpha_m \rangle)$ .

Function  $insideGrid$  verifies if an  $m$ -dimensional box  $B$  is contained in a grid box of an  $m$ -dimensional grid. Its pseudo code is presented in algorithm 17.

---

<sup>1</sup>Only the relation  $\leq$  is being considered in inequalities.

---

**Algorithm 17:**  $insideGrid(I_1 \times \dots \times I_m, \langle \alpha_1, \dots, \alpha_m \rangle)$

<p><b>Input:</b> <math>I_1 \times \dots \times I_m</math>: box; <math>\langle \alpha_1, \dots, \alpha_l \rangle</math>: tuple of doubles;  <b>Output:</b> <math>flag</math>: boolean;</p> <pre> 1 <math>flag \leftarrow true</math>; 2 <b>for</b> <math>i \leftarrow 1</math> <b>to</b> <math>m</math> <b>do</b> 3   <math>flag = flag \wedge (ceil(\overline{I}_i/\alpha_i) - floor(\underline{I}_i/\alpha_i) \leq 1)</math>; 4 <b>end</b> 5 <b>return</b> <math>flag</math>;</pre>
--

### Eligible Predicate

The *eligible* predicate verifies if a box  $B$  is eligible to split. The default parametrization,  $eligible_\varepsilon$ , requires that the width of box  $B$  is largest than a given  $\varepsilon$ , i.e.  $eligible_\varepsilon(B) \equiv wid(B) > \varepsilon$ .

### Eligible Grid

In section 5.4 a grid oriented implementation for the *eligible* predicate is required,  $eligible_\alpha$ , such that  $eligible_\alpha(B) \equiv \neg insideGrid(\Pi_{Z_{idx}}(B), \langle \alpha_1, \dots, \alpha_m \rangle)$ .

### Stop Predicate

The *stop* predicate verifies if it is possible to interrupt the *cReasoning* algorithm based on the current state of the joint box cover.

In this thesis most of the algorithms that use *cReasoning* compute an enclosure of a quantity based on the boxes of a joint box cover (e.g., enclosure for the probability of an event, enclosure for the integral of a function over a region, enclosure for the volume of a region, enclosure for the reliability of a design, etc.). Usually the goal is to stop when the computed enclosure satisfies a given criterion, namely:

- $stop_\delta$  verifies if the enclosure has reached the desired accuracy,  $\delta$ ;
- $stop_\delta \geq$  verifies if the enclosure has reached the desired accuracy,  $\delta$  or if its upper bound is less than a given threshold,  $tol$ .

Other alternative to interrupt the loop in *cReasoning* is to stop after a predefined amount of time  $T$ , resulting in the  $stop_T$  predicate.

## Split Function

Function *split* is any generic technique that splits a box in two or more sub-boxes. Its default parametrization, *split*<sub>2</sub>, splits the box largest interval in its midpoint, resulting in two sub-boxes.

## Split Grid

In section 5.4 a grid oriented implementation for the *split* function is required, *split*<sub>α</sub>. Algorithm 18 presents the pseudo-code of such function, that splits an *n*-dimensional box in a grid point of a chosen grid interval, wrt a grid with spacings  $\langle \alpha_1, \dots, \alpha_m \rangle$ , producing a list with two sub boxes (lines 11 – 16). The chosen interval is the one, among the *m* dimensions, that has largest width and simultaneously spans for more than one unit grid interval in that dimension (lines 1 – 10).

**Algorithm 18:**  $split_\alpha(B, \langle x_1, \dots, x_m \rangle, \langle \alpha_1, \dots, \alpha_m \rangle)$

<p><b>Input:</b> <i>B</i>: box; <math>\langle x_1, \dots, x_m \rangle</math>: tuple of variables; <math>\langle \alpha_1, \dots, \alpha_m \rangle</math>: tuple of doubles;  <b>Output:</b> <math>\{B_l, B_r\}</math>: set of boxes;</p> <pre> 1 <math>w \leftarrow 0;</math>      <math>j \leftarrow 0;</math> 2 <b>for</b> <math>i \leftarrow 1</math> <b>to</b> <math>m</math> <b>do</b> 3   <math>I_i \leftarrow \Pi_i(B);</math> 4   <b>if</b> (<math>wid(I_i) &gt; w</math>) <b>then</b> 5     <b>if</b> (<math>ceil(\overline{I}_i/\alpha_i) - floor(\underline{I}_i/\alpha_i) &gt; 1</math>) <b>then</b> 6       <math>j \leftarrow i;</math> 7       <math>w \leftarrow wid(I_i);</math> 8     <b>end</b> 9   <b>end</b> 10 <b>end</b> 11 <math>I_j \leftarrow \Pi_j(B);</math> 12 <math>l_g \leftarrow floor(mid(I_j)/\alpha_j)\alpha_j;</math>      <math>r_g \leftarrow ceil(mid(I_j)/\alpha_j)\alpha_j;</math> 13 <b>if</b> (<math>mid(I_j) - l_g \leq r_g - mid(I_j)</math>) <b>then</b> <math>p \leftarrow l_g;</math> <b>else</b> <math>p \leftarrow r_g;</math> 14 <math>B_l \leftarrow B;</math>      <math>\Pi_j(B_l) \leftarrow [I_j, p];</math> 15 <math>B_r \leftarrow B;</math>      <math>\Pi_j(B_r) \leftarrow [p, \overline{I}_j];</math> 16 <b>return</b> <math>\{B_l, B_r\};</math> </pre>
--

## Order Criterion

The *order* criterion imposes an order on the boxes of outer covers, specifying how they are selected for processing. Usually it gives preference to boxes with highest contribution to reach the condition imposed by the *stop* predicate. Namely:

- $order_P$ , orders the boxes by decreasing order of the width of their probability enclosure;
- $order_W$  orders the boxes by decreasing order of their width;
- $order_{LIFO}$  induces the behavior of a *LIFO* data structure to the outer cover;
- $order_{\downarrow}$  induces a depth first search.

## A.2 Algorithm Convergence

This section discusses the convergence of algorithm *cReasoning* (algorithm 3) assuming that it is implemented with an infinite precision interval arithmetic.

The proof is based on a similar one presented in [48]. So, we directly use some results from such paper, as the following proposition, which states that a convergent inclusion function  $[g]$  of  $g$  allows computing arbitrarily sharp enclosures of the set  $\{x \in D : g(x) = 0\}$ , where  $D \subseteq \mathbb{R}^n$ .

**Proposition A.1** *Consider a continuous function  $g : \mathbb{R}^n \rightarrow \mathbb{R}$  and an inclusion function  $[g]$  of  $g$  that is convergent inside a bounded  $D$ . Let  $\Omega_0 = \{x \in D : g(x) = 0\}$  and consider an arbitrary set  $\Omega_0^+ \subseteq \mathbb{R}^n$  such that  $\Omega_0 \subseteq \text{int}(\Omega_0^+)$ , where  $\text{int}(E)$  is the interior of  $E$ . Then there exists  $\varepsilon > 0$  such that for all  $B \subseteq D$  we have  $\text{wid}(B) \leq \varepsilon$  and  $0 \in [g](B)$  imply  $B \subseteq \Omega_0^+$ .*

**Proof.** For a proof see [48]. ■

In the following consider:

- CCSP  $\langle X, D, C \rangle$ , where  $D \subseteq \mathbb{R}^n$  and  $C = \{(s_1, \rho_1), \dots, (s_m, \rho_m)\}$ , with  $\rho_i = \{x \in D : g_i(x) \diamond_i 0\}$ ;

- $\mathcal{F}_{\boxplus_k} = \langle \mathcal{F}_{\square_k}, \mathcal{F}_{\blacksquare_k} \rangle$  to be the joint box cover computed at iteration  $k$  of the while loop in *cReasoning*;
- $\Delta_k \mathcal{F} = \mathcal{F}_k^+ \setminus \mathcal{F}_k^-$ ;
- $\varepsilon_k = \max_{B \in \mathcal{F}_{\square_k} \setminus \mathcal{F}_{\blacksquare_k}} \text{wid}(B)$ ;
- $\mathcal{F}_{\boxplus_0} = \langle \{D\}, \emptyset \rangle$  to be the input joint box cover of *cReasoning*;
- *cReasoning* $_{\infty}$  to be parameterized as follows:
  - the *stop* predicate returns *false*;
  - the *inner* predicate is *inner<sub>d</sub>* (defined in section A.1);
  - the conjunction of the *order* criterion and the *eligible* predicate imposes a fair selection strategy wrt boundary boxes;
  - the *split* function is fair.

Intuitively, the fairness of the selection strategy guarantees that when  $k$  approaches infinity, all eligible boxes will have been split an infinite number of times. The fairness of the split strategy furthermore guarantees that when the number of splits of a box approaches infinity its width approaches zero (*split<sub>2</sub>*, defined in section A.1, is fair).

The next property states that the width of the largest boundary box approaches zero as  $k$  approaches infinity.

**Property A.1** Consider a sequence  $(\mathcal{F}_{\boxplus_k})_{k \in \mathbb{N}}$  computed by *cReasoning* $_{\infty}$  such that  $\mathcal{F}_{\boxplus_k} = \text{crStep}(\mathcal{F}_{\boxplus_{k-1}}, C, \text{split}, \text{inner}, \text{true}, \text{order})$ . Then  $\lim_{k \rightarrow \infty} \varepsilon_k = 0$ .

**Proof.** [Sketch] The fairness of the selection and of the split strategies guarantees that, for all boxes in  $\mathcal{F}_{\square_k} \setminus \mathcal{F}_{\blacksquare_k}$ , when  $k$  approaches infinity their width approaches zero and so does  $\varepsilon_k$ . For a formal definition of fairness of the split and selection strategies and a proof of this property see [48]. ■

Given a function  $g$  whose set of roots is the union of the constraint boundaries, the next property states that, for every joint box cover  $\mathcal{F}_{\boxplus_k}$ , the evaluation of the inclusion function  $[g]$  over a boundary box results in an interval that contains zero.

**Property A.2** Let  $g : D \rightarrow \mathbb{R}$  be a continuous function defined as  $g(x) = g_1(x) \times \cdots \times g_m(x)$  such that  $\Omega_0 = \{x \in D : g(x) = 0\} = \bigcup_{i=1}^m \{x \in D : g_i(x) = 0\}$ . Given the properties of narrowing operators and of the  $inner_d$  predicate:

$$\forall_{k>0} \forall_{B \in \mathcal{F}_{\square_k} \setminus \mathcal{F}_{\blacksquare_k}} 0 \in [g](B)$$

**Proof.** Let us start by notice that lines 9 and 10 of *crStep* (algorithm 2) guarantee that boxes added to  $\mathcal{F}_{\square_k}$  are the ones not removed by  $\mathcal{CPA}$  algorithm and those added to  $\mathcal{F}_{\blacksquare_k}$  are the ones identified as inner boxes by the *inner* predicate (consequently such boxes do not belong to  $\mathcal{F}_{\square_k} \setminus \mathcal{F}_{\blacksquare_k}$ ).

The proof is carried out by contradiction. Assume that exists  $k$  and  $B \in \mathcal{F}_{\square_k} \setminus \mathcal{F}_{\blacksquare_k}$  such that  $0 \notin [g](B)$ . In that case one of the following is true:

- $\forall_i \inf([g_i](B)) > 0$ ;
- $\forall_i \sup([g_i](B)) < 0$ .

In the first case, by the properties of the narrowing operators (see section 2.3.1),  $\mathcal{CPA}$  algorithm would have removed box  $B$ . In the second case, by the properties of the  $inner_d$  predicate, if all constraints are inequalities  $B$  would have been added to  $\mathcal{F}_{\blacksquare_k}$ . Otherwise, if some constraint is an equation,  $B$  would have been removed by  $\mathcal{CPA}$  algorithm. In either case  $B \notin \mathcal{F}_{\square_k} \setminus \mathcal{F}_{\blacksquare_k}$  which is a contradiction. ■

**Property A.3 (Convergence)** Consider a sequence  $(\mathcal{F}_{\boxplus_k})_{k \in \mathbb{N}}$  computed by  $cReasoning_\infty$  such that  $\mathcal{F}_{\boxplus_k} = crStep(\mathcal{F}_{\boxplus_{k-1}}, C, split, inner, true, order)$ . Then  $\lim_{k \rightarrow \infty} vol(\Delta_k \mathcal{F}) = 0$ .

**Proof.** Notice that  $\Omega_0 = \bigcup_{i=1}^m \{x \in D : g_i(x) = 0\}$  is a null-volume set since it is the union of  $m$  null-volume sets. So, by definition of a null-volume set, for all  $\delta > 0$  there exists a set of boxes  $S$  such that  $vol(\bigcup S) \leq \frac{1}{2^n} \delta$  and  $\Omega_0 \subseteq \bigcup S$ . Now define

$$S^+ = \{mid(B) + 2(B - mid(B)) : B \in S\},$$

informally each box of  $S$  sees each of its dimension inflated by a ratio of 2. Thus we now have  $vol(\bigcup S^+) \leq \delta$  and  $\Omega_0 \subseteq int(\bigcup S^+)$ . By proposition A.1 there exists  $\varepsilon'_\delta$  such that for all  $B \subseteq D$ ,  $wid(B) \leq \varepsilon'_\delta$  and  $0 \in [g](B)$  implies  $B \subseteq \bigcup S^+$ . Since the boxes  $B \in \mathcal{F}_{\square_k} \setminus \mathcal{F}_{\blacksquare_k}$  satisfy  $B \subseteq D$ ,  $wid(B) \leq \varepsilon_k$  and, by property A.2,  $0 \in [g](B)$  we obtain that  $\varepsilon_k \leq \varepsilon'_\delta$  implies  $\Delta_k \mathcal{F} \subseteq (\bigcup S^+)$  and thus  $vol(\Delta_k \mathcal{F}) \leq \delta$ . As this holds for an arbitrary  $\delta > 0$  and  $\varepsilon_k$  converges to zero, we have that  $vol(\Delta_k \mathcal{F})$  also converges to zero. ■

## APPENDIX B

# Integration with Taylor Models

## B.1 Proof of Property 3.16

The following properties, used next in the proof of property 3.16, relate the width of some operations on intervals.

**Property B.1 (Width of the Sum of Intervals)** *Given two intervals  $I_1 = [a, b]$  and  $I_2 = [c, d]$  then  $\text{wid}(I_1 + I_2) = \text{wid}(I_1) + \text{wid}(I_2)$ .*

**Proof.** By definition  $\text{wid}([a, b] + [c, d]) = \text{wid}([a + c, b + d]) = (b + d) - (a + c) = -a + b - c + d$ . Since  $\text{wid}([a, b]) + \text{wid}([c, d]) = (b - a) + (d - c) = -a + b - c + d$ , property B.1 holds. ■

**Property B.2 (Width of the Product of Intervals)** *When  $I_1 = [-a, a]$  and  $I_2 = [-b, b]$  or  $I_2 = [0, b]$  then  $\text{wid}(I_1 \cdot I_2) \leq \text{wid}(I_1)\text{wid}(I_2)$ .*

**Proof.** In the first case  $\text{wid}(I_1 \cdot [-b, b]) = \text{wid}([-ab, ab]) = 2ab$  and  $\text{wid}(I_1)\text{wid}([-b, b]) = 2a \cdot 2b = 4ab$ . In the second,  $\text{wid}(I_1 \cdot [0, b]) = \text{wid}([-ab, ab]) = 2ab$  and  $\text{wid}(I_1)\text{wid}([0, b]) = 2ab$ . Hence, property B.2 holds in both cases. ■

**Property B.3 (Width of a Power of a Centered Interval)** *When  $I = [-a, a]$  is an interval centered at 0 and  $j$  is a positive integer then  $\text{wid}([-a, a]^j) \leq (\text{wid}([-a, a]))^j$ .*

**Proof.** Since

$$\text{wid}([-a, a]^j) = \begin{cases} \text{wid}([0, a^j]) = a^j & \text{if } \text{even}(j) \\ \text{wid}([-a^j, a^j]) = 2a^j & \text{if } \text{odd}(j) \end{cases}$$

and  $(\text{wid}([-a, a]))^j = (2a)^j = 2^j a^j$ , property B.3 holds. ■

**Property B.4 (Width of the Product of Powers of Centered Intervals)**

Consider a box  $B \in \mathbb{IR}^m$ , its midpoint  $\tilde{x}$  and a multi-index  $\alpha = \langle \alpha_1, \dots, \alpha_m \rangle$  where  $|\alpha| = n$ , then  $\text{wid}((B - \tilde{x})^\alpha) \leq (\text{wid}(B))^n$ .

**Proof.** All the intervals of box  $(B - \tilde{x})$  are centered at 0, i.e.  $(B - \tilde{x}) = [-a_1, a_1] \times \dots \times [-a_m, a_m]$  and each interval as the same width as its counterpart in box  $B$ .

Since  $(B - \tilde{x}) = [-a_1, a_1] \times \dots \times [-a_m, a_m]$  we have that

$$\begin{aligned} \text{wid}((B - \tilde{x})^\alpha) &= \text{wid}([-a_1, a_1]^{\alpha_1} \dots [-a_m, a_m]^{\alpha_m}) && \text{multi-index power} \\ &\leq \text{wid}([-a_1, a_1]^{\alpha_1}) \dots \text{wid}([-a_m, a_m]^{\alpha_m}) && \text{property B.2} \\ &\leq (\text{wid}([-a_1, a_1]))^{\alpha_1} \dots (\text{wid}([-a_m, a_m]))^{\alpha_m} && \text{property B.3} \\ &\leq (\text{wid}(B))^{\alpha_1} \dots (\text{wid}(B))^{\alpha_m} && \text{definition of box width} \\ &= (\text{wid}(B))^n && |\alpha| = n \end{aligned}$$

and hence property B.4 holds. ■

Now follows the proof of property 3.16, restated here for completeness.

**Property B.5 (Order of Convergence)** *The quadrature computed as in lemma 3.1 has an order of convergence  $n + 2 + m$  in a box  $B \in \mathbb{IR}^m$ , when an  $n$ -order Taylor model is used.*

**Proof.** Property 3.16 states that, for all  $B' \subseteq B$  (see definition 2.9):

$$\text{wid} \left( \int_{B'} p(x) dx + R \text{vol}(B') \right) - \text{wid} \left( \int_{B'} f(x) dx \right) \leq k (\text{wid}(B'))^{n+2+m} \quad (\text{B.1})$$

In fact, the left hand side can be expanded to

$$wid\left(\int_{B'} p(x)dx\right) + wid(R)vol(B') - wid\left(\int_{B'} f(x)dx\right) \quad (\text{B.2})$$

Since the result of an integral over a box is a scalar we have that

$$wid\left(\int_{B'} p(x)dx\right) = 0 \quad \text{and} \quad wid\left(\int_{B'} f(x)dx\right) = 0$$

Hence (B.2) can be rewritten as  $wid(R)vol(B')$ .

By definition of volume and width of a box we have the inequality  $vol(B') \leq (wid(B'))^m$ .

We will now focus on  $wid(R)$ . From (3.2),  $R = \sum_{|\alpha|=n+1} ([r_\alpha](B') - c_\alpha)(B' - \tilde{x})^\alpha$  hence

$$wid(R) = wid\left(\sum_{|\alpha|=n+1} ([r_\alpha](B') - c_\alpha)(B' - \tilde{x})^\alpha\right) \quad (\text{B.3})$$

$$= \sum_{|\alpha|=n+1} wid([r_\alpha](B') - c_\alpha)(B' - \tilde{x})^\alpha \quad (\text{B.4})$$

given property B.1.

Since  $([r_\alpha](B') - c_\alpha)$  is centered at 0 and  $(B' - \tilde{x})^\alpha$  is centered at 0 if  $\exists_{\alpha_i \in \alpha} \text{odd}(\alpha_i)$  or has the form  $[0, a]$  otherwise, then, by property B.2

$$wid([r_\alpha](B') - c_\alpha)(B' - \tilde{x})^\alpha \leq wid([r_\alpha](B') - c_\alpha)wid((B' - \tilde{x})^\alpha)$$

Hence, from (B.4),

$$\begin{aligned} wid(R) &\leq \sum_{|\alpha|=n+1} wid([r_\alpha](B') - c_\alpha)wid((B' - \tilde{x})^\alpha) \\ &= \sum_{|\alpha|=n+1} wid([r_\alpha](B'))wid((B' - \tilde{x})^\alpha) \quad c_\alpha \text{ is a constant, } wid(c_\alpha) = 0 \\ &\leq \sum_{|\alpha|=n+1} k_\alpha wid(B')wid((B' - \tilde{x})^\alpha) \quad [r_\alpha](B') \text{ as linear convergence} \\ &\leq \sum_{|\alpha|=n+1} k_\alpha wid(B')(wid(B'))^{n+1} \quad \text{property B.4} \\ &= \sum_{|\alpha|=n+1} k_\alpha (wid(B'))^{n+2} \end{aligned}$$

$$\begin{aligned}
 &\leq l \max(k_\alpha)(\text{wid}(B'))^{n+2} && l = \text{number of terms in } \sum \\
 &= k(\text{wid}(B'))^{n+2} && \text{with } k = l \max(k_\alpha)
 \end{aligned}$$

Consequently  $\text{wid}(R)\text{vol}(B') \leq k(\text{wid}(B'))^{n+2}(\text{wid}(B'))^m = k(\text{wid}(B'))^{n+2+m}$  and so inequality (B.1) holds.  $\blacksquare$

## B.2 Numerical Computations of Example 3.2

Consider function  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ , that describes the standard bivariate normal PDF, with correlation coefficient  $\rho = 0.5$ , given by:

$$f(x, y) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right)$$

We use the method based on Taylor models to compute the enclosure for the quadrature of this function in the box  $B = [0, 0.5] \times [0, 0.5]$  (see Lemma 3.1), adopting the Taylor model of order  $n = 2$  around the midpoint of  $B$ ,  $\tilde{x} = \langle 0.25, 0.25 \rangle$ .

We start by calculating the remainder  $R$ . For that purpose we calculate  $[r_\alpha](B)$ , for each pair where  $|\alpha| = 3$ . These pairs are  $\alpha_{30} = \langle 3, 0 \rangle$ ,  $\alpha_{21} = \langle 2, 1 \rangle$ ,  $\alpha_{12} = \langle 1, 2 \rangle$  and  $\alpha_{03} = \langle 0, 3 \rangle$ .

In the following formulas  $g(x, y) = \frac{8}{27\sqrt{3}\pi} \exp(-\frac{2}{3}(x^2 - xy + y^2))$ .

$$\begin{aligned}
 [r_{\alpha_{30}}](B) &= \frac{1}{3!} \left[ \frac{\partial^3 f}{\partial x^3} \right] (B) = \left[ \frac{g(x, y)}{6} (-8x^3 + y^3 + 12x^2y - 6xy^2 + 18x - 9y) \right] (B) \\
 [r_{\alpha_{21}}](B) &= \frac{1}{2!} \left[ \frac{\partial^3 f}{\partial x^2 \partial y} \right] (B) = \left[ \frac{g(x, y)}{2} (4x^3 - 2y^3 - 12x^2y + 9xy^2 - 9x + 9y) \right] (B) \\
 [r_{\alpha_{12}}](B) &= \frac{1}{2!} \left[ \frac{\partial^3 f}{\partial x \partial y^2} \right] (B) = \left[ \frac{g(x, y)}{2} (-2x^3 + 4y^3 + 9x^2y - 12xy^2 + 9x - 9y) \right] (B) \\
 [r_{\alpha_{03}}](B) &= \frac{1}{3!} \left[ \frac{\partial^3 f}{\partial y^3} \right] (B) = \left[ \frac{g(x, y)}{6} (x^3 - 8y^3 - 6x^2y + 12xy^2 - 9x + 18y) \right] (B)
 \end{aligned}$$

For each  $\alpha$ , we have that  $c_\alpha = \text{mid}([r_\alpha](B))$ .

Let us now calculate  $(B - \langle [0.25], [0.25] \rangle)^\alpha$ , for each  $\alpha$ .

$$\begin{aligned}
 (B - \langle [0.25], [0.25] \rangle)^{\alpha_{30}} &= ([0, 0.5] - [0.25])^3 ([0, 0.5] - [0.25])^0 \\
 &= [-0.25, 0.25]^3 [-0.25, 0.25]^0 = [-1/64, 1/64] [1] \\
 &= [-1/64, 1/64] \\
 (B - \langle [0.25], [0.25] \rangle)^{\alpha_{21}} &= ([0, 0.5] - [0.25])^2 ([0, 0.5] - [0.25])^1 \\
 &= [-0.25, 0.25]^2 [-0.25, 0.25]^1 = [1/16] [-0.25, 0.25] \\
 &= [-1/64, 1/64]
 \end{aligned}$$

It is easy to confirm that  $(B - \tilde{x})^{\alpha_{12}} = (B - \tilde{x})^{\alpha_{21}}$  and  $(B - \tilde{x})^{\alpha_{03}} = (B - \tilde{x})^{\alpha_{30}}$ . So

$$\begin{aligned}
 R = [-1/64, 1/64] &([r_{\alpha_{30}}](B) - c_{\alpha_{30}} + [r_{\alpha_{21}}](B) - c_{\alpha_{21}} + \\
 &[r_{\alpha_{12}}](B) - c_{\alpha_{12}} + [r_{\alpha_{03}}](B) - c_{\alpha_{03}})
 \end{aligned}$$

Using interval analysis to compute  $R$  we obtain  $0.0090460851600035 \times [-1, 1]$ . Since  $\text{vol}(B) = 0.25$ , then  $R \text{vol}(B) = 0.0022615212900009 \times [-1, 1]$ .

Now we compute  $\int_B p(x) dx$  using property 3.14, by calculating the following terms:

$$\begin{aligned}
 T_1 &= \text{vol}(B) f(\tilde{x}) = 0.25 \times 0.176276287962736 \binom{6}{4} = 0.044069071990684 \binom{2}{1} \\
 T_2 &= \sum_{|\alpha|=1}^2 K_\alpha \int_B (x - \tilde{x})^\alpha dx \\
 T_3 &= \sum_{|\alpha|=3} c_\alpha \int_B (x - \tilde{x})^\alpha dx = 0
 \end{aligned}$$

The term  $T_3$  is 0 because when  $|\alpha| = 3$  all the possible pairs for  $\alpha$  ( $\langle 3, 0 \rangle$ ,  $\langle 2, 1 \rangle$ ,  $\langle 1, 2 \rangle$  and  $\langle 0, 3 \rangle$ ) contain an odd number (see property 3.15).

In the term  $T_2$ , when  $|\alpha| = 1$  all the possible pairs for  $\alpha$  ( $\langle 1, 0 \rangle$  and  $\langle 0, 1 \rangle$ ) contain an odd number, so  $|\alpha| = 1$  can be ignored from the sum. When  $|\alpha| = 2$  the possible pairs for  $\alpha$  are  $\langle 2, 0 \rangle$ ,  $\langle 1, 1 \rangle$  and  $\langle 0, 2 \rangle$ , and  $\langle 1, 1 \rangle$  can be ignored since it contains odd

numbers. So, the term  $T_2$  simplifies to

$$\begin{aligned} T_2 &= \frac{1}{2!} \left[ \frac{\partial^2 f}{\partial x^2} \right] (\tilde{x}) \frac{(0.5)^3 (0.5)^1}{2^2 3} + \frac{1}{2!} \left[ \frac{\partial^2 f}{\partial y^2} \right] (\tilde{x}) \frac{(0.5)^1 (0.5)^3}{2^0 1} \\ &= \frac{1}{2^7 3} \left( \left[ \frac{\partial^2 f}{\partial x^2} \right] (\tilde{x}) + \left[ \frac{\partial^2 f}{\partial y^2} \right] (\tilde{x}) \right) \end{aligned}$$

By this calculations we obtain

$$\begin{aligned} \int_B p(x) dx &= T_1 + T_2 \\ &= 0.044069071990684 \binom{2}{1} - 0.001198637953450 \binom{4}{3} \\ &= 0.042870434037233 \binom{9}{7} \end{aligned}$$

We can notice that the polynomial part is *almost* a scalar (the intervals are there just to guarantee bounds for floating-point rounding errors).

Finally we compute the desired enclosure for the quadrature of  $f$  in  $B$ :

$$\begin{aligned} \int_B f(x) dx &\in \int_B p(x) dx + R \operatorname{vol}(B) = \\ &0.042870434037233 \binom{9}{7} + 0.0022615212900009 \times [-1, 1] = \\ &[0.0406089127472329, 0.0451319553272347]. \end{aligned}$$

## References

- [1] Gaol, Not Just Another Interval Library. [www.sourceforge.net/projects/gaol](http://www.sourceforge.net/projects/gaol). 18, 107
- [2] In G. D. Cooman, D. Ruan, and E. Kerre, editors, *Foundations and Applications of Possibility Theory*, pages 88–98. World Scientific, 1995. 45
- [3] Remote sensing of inherent optical properties: Fundamentals, tests of algorithms, and applications. IOCCG Report NUMBER 5, 2006. 177
- [4] H. Agarwal, S. Gano, V. Perez, C. Mozumder, J. Renaud, and L. Watson. Homotopy methods for constraint relaxation in unilevel reliability based design optimization. *Engineering Optimization*, 41(6):593–607, 2009. 219
- [5] H. Agarwal, J. Lee, L. Watson, and J. Renaud. A unilevel method for reliability based design optimization. In *45th Structures, Structural Dynamics, and Materials Conf.*, 2004. 196
- [6] M. Alrefaei and H. Abdul-Rahman. An adaptive Monte Carlo integration algorithm with general division approach. *Math. Comput. Simul.*, 79:49–59, October 2008. 183
- [7] I. Araya, B. Neveu, and G. Trombettoni. Exploiting common subexpressions in numerical CSPs. In *CP*, pages 342–357, 2008. 34
- [8] I. Araya, G. Trombettoni, and B. Neveu. Filtering numerical CSPs using well-constrained subsystems. In *CP*, pages 158–172, 2009. 35
- [9] I. Araya, G. Trombettoni, and B. Neveu. Exploiting monotonicity in interval constraint propagation. In *AAAI*, 2010. 34

## REFERENCES

---

- [10] F. Barranco-Cicilia, E. C.-P. de Lima, and L. Sudati-Sagrilo. Structural Reliability Analysis of Limit State Functions With Multiple Design Points Using Evolutionary Strategies. *Ing. invest. y tecnol.*, 10:87 – 97, 2009. [195](#)
- [11] D. M. Bates and D. G. Watts. *Nonlinear Regression: Iterative Estimation and Linear Approximations*, pages 32–65. Wiley, 1988. [155](#), [157](#), [178](#)
- [12] Y. Ben-Haim and F. Elishako. *Convex models of uncertainty in applied mechanics*. Elsevier Science, 1990. [45](#)
- [13] F. Benhamou and F. Goualard. Universally quantified interval constraints. In *Principles and Practice of Constraint Programming*, pages 67–82, 2000. [80](#), [188](#)
- [14] F. Benhamou, F. Goualard, L. Granvilliers, and J. Puget. Revising hull and box consistency. In *Proceedings of the 1999 international conference on Logic programming*, pages 230–244, Cambridge, MA, USA, 1999. Massachusetts Institute of Technology. [33](#), [34](#), [107](#)
- [15] F. Benhamou, D. McAllester, and P. van Hentenryck. CLP(intervals) revisited. In *ISLP*, pages 124–138. MIT Press, 1994. [32](#), [33](#)
- [16] M. Berz and G. Hoffstätter. Computation and application of Taylor polynomials with interval remainder bounds. *Reliable Computing*, 4(1):83–97, 1998. [23](#)
- [17] M. Berz and K. Makino. New methods for high-dimensional verified quadrature. *Reliable Computing*, 5:13–22, 1999. [68](#), [71](#)
- [18] C. H. Bischof, P. Hovland, and B. Norris. On the implementation of automatic differentiation tools. *Higher-Order and Symbolic Computation*, 21(3):311–331, 2008. [70](#)
- [19] S. Bistarelli, U. Montanari, and F. Rossi. Constraint solving over semirings. In *In Proc. IJCAI95*, pages 624–630. Morgan, 1995. [79](#)
- [20] K. Breitung. Asymptotic Approximation for Multinormal Integrals. *J. Eng. Mech.*, 110(3):357 – 366, 1984. [194](#)
- [21] H. Brönnimann, G. Melquiond, and S. Pion. The design of the Boost interval arithmetic library. *Theor. Comput. Sci.*, 351:111–118, February 2006. [18](#)

- 
- [22] R. E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7(-1):1–49, 1998. [74](#)
- [23] E. Carvalho, J. Cruz, and P. Barahona. Probabilistic reasoning with continuous constraints. *AIP Conference Proceedings*, 936(1):105–108, 2007. [225](#)
- [24] E. Carvalho, J. Cruz, and P. Barahona. Probabilistic continuous constraint satisfaction problems. In *ICTAI (2)*, pages 155–162, 2008. [225](#)
- [25] E. Carvalho, J. Cruz, and P. Barahona. Probabilistic reasoning for inverse problems. In *Advances in Soft Computing*, volume 46, pages 115–128. Springer-Verlag, 2008. [229](#)
- [26] E. Carvalho, J. Cruz, and P. Barahona. Probabilistic constraints for reliability problems. In *Proceedings of the 25th Annual ACM Symposium on Applied Computing (SAC 2010)*, Sierre, Switzerland, 2010. ACM. [229](#)
- [27] E. Carvalho, J. Cruz, and P. Barahona. Reasoning with uncertainty in continuous domains. In *Advances in Intelligent and Soft Computing*. Springer-Verlag, 2010. [229](#)
- [28] E. Carvalho, J. Cruz, and P. Barahona. Probabilistic Constraints for Nonlinear Inverse Problems. *Accepted for publication in Constraints*, 2012. [229](#)
- [29] G. Chabert and L. Jaulin. Hull consistency under monotonicity. In I. P. Gent, editor, *Principles and Practice of Constraint Programming - CP 2009*, volume 5732 of *Lecture Notes in Computer Science*, pages 188–195. Springer, 2009. [34](#)
- [30] I. Charpentier and J. Utke. Fast higher-order derivative tensors with Rapsodia. *Optimization Methods Software*, 24(1):1–14, 2009. [70](#)
- [31] X. Chen, T. Hasselman, and D. Neill. Reliability-based structural design optimization for practical applications. In *38th Structures, Structural Dynamics, and Materials Conf.*, 1997. [196](#)
- [32] G. Cheng, L. Xu, and L. Jiang. A sequential approximate programming strategy for reliability-based structural optimization. *Computers and Structures*, 84(21):1353 – 1367, 2006. [219](#)

## REFERENCES

---

- [33] S. Choi, R. Grandhi, and R. Canfield. *Reliability-Based Structural Design*. Springer, 2010. [200](#), [201](#)
- [34] J. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2):125–149, 1987. [33](#)
- [35] H. Collavizza, F. Delobel, and M. Rueher. A note on partial consistencies over continuous domains. *Lecture Notes in Computer Science*, 1520:147–161, 1998. [30](#)
- [36] H. Collavizza, F. Delobel, and M. Rueher. Comparing partial consistencies. *Reliable Computing*, 5(3):213–228, 1999. [32](#)
- [37] G. Corliss and L. Rall. Adaptive, self-validating numerical quadrature. *SIAM J. Sci. Stat. Comput.*, 8:831–847, 1987. [67](#)
- [38] K. Deb, D. Padmanabhan, S. Gupta, and A. Mall. Handling uncertainties through reliability-based optimization using evolutionary algorithms, 2006. [191](#), [196](#), [209](#)
- [39] O. Ditlevsen. Narrow reliability bounds for structural system. *J. St. Mech.*, 4:431–439, 1979. [196](#)
- [40] X. Du and W. Chen. Sequential optimization and reliability assessment method for efficient probabilistic design. In *Probabilistic Design, ASME Design Engineering Technical Conferences*, pages 225–233, 2002. [196](#)
- [41] D. Dubois, H. Fargier, and H. Prade. The calculus of fuzzy restrictions as a basis for flexible constraint satisfaction. In *Fuzzy Systems, 1993., Second IEEE International Conference on*, volume 2, pages 1131–1136, 1993. [79](#)
- [42] M. Eldred, B. Bichon, and B. Adams. Overview of Reliability Analysis and Design Capabilities in DAKOTA. In *Workshop on Reliable Engineering Computing*, pages 1–26. 2006. [194](#)
- [43] H. Fargier and J. Lang. Uncertainty in constraint satisfaction problems: a probabilistic approach. In *Proc. of ECSQARU*, pages 97–104. Springer-Verlag LNCS 747, 1993. [79](#)
- [44] H. Fargier, J. Lang, R. Martin-Clouaire, and T. Schiex. A constraint satisfaction framework for decision under uncertainty. In *In Proc. of the 11th Int. Conf. on Uncertainty in Artificial Intelligence*, pages 175–180, 1995. [79](#)

- 
- [45] S. Ferson, V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz. Constructing Probability Boxes and Dempster-Shafer Structures. *SAND2002-4015*, (January):1–143, 2003. [45](#)
- [46] B. Fiessler, H.-J. Neumann, and R. Rackwitz. Quadratic limit states in structural reliability. *J. Engrg. Mech. Div.*, 105:661–676, 1979. [192](#)
- [47] H. Goel, J. Grievink, P. Herder, and M. Weijnen. Integrating reliability optimization into chemical process synthesis. *Reliability Engineering and System Safety*, 78(3):247–258, 2002. [187](#)
- [48] A. Goldsztejn, J. Cruz, and E. Carvalho. Convergence analysis and adaptive strategy for the certified quadrature over a set defined by inequalities. *Submitted to Journal of Computational and Applied Mathematics*, 2011. [70](#), [99](#), [226](#), [237](#), [238](#)
- [49] A. Goldsztejn and F. Goualard. Box consistency through adaptive shaving. In *SAC*, pages 2049–2054, 2010. [34](#)
- [50] A. Goldsztejn and L. Granvilliers. A new framework for sharp and efficient resolution of ncp with manifolds of solutions. *Constraints*, 15(2):190–212, 2010. [35](#)
- [51] L. Granvilliers. RealPaver User’s Manual, version 0.4, 2004. [pagesperso.lina.univ-nantes.fr/~granvilliers-l/realpaver](http://pagesperso.lina.univ-nantes.fr/~granvilliers-l/realpaver). [107](#)
- [52] L. Granvilliers and F. Benhamou. Algorithm 852: Realpaver: an interval solver using constraint satisfaction techniques. *ACM Trans. Math. Softw.*, 32(1):138–156, 2006. [107](#), [228](#)
- [53] L. Granvilliers, J. Cruz, and P. Barahona. Parameter estimation using interval computations. *SIAM J. Scientific Computing*, 26(2):591–612, 2004. [155](#), [158](#)
- [54] A. Griewank, D. Juedes, and J. Utke. Algorithm 755: ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM Transactions on Mathematical Software*, 22(2):131–167, 1996. [70](#)

## REFERENCES

---

- [55] A. Halder and S. Mahadevan. *Probability, Reliability and Statistical Methods in Engineering Design*. Wiley, 1999. [196](#)
- [56] J. Hammersley and D. Handscomb. *Monte Carlo Methods*. Methuen, London, 1964. [72](#)
- [57] E. Hansen. A globally convergent interval method for computing and bounding real roots. *BIT Numerical Mathematics*, 18:415–424, 1978. [25](#)
- [58] E. Hansen. *Global Optimization Using Interval Analysis*. Marcel Dekker, New York, 1992. [25](#)
- [59] E. Hansen and S. Sengupta. Bounding solutions of systems of equations using interval analysis. *Bit*, 21:203–221, 1981. [26](#)
- [60] A. M. Hasofer and N. C. Lind. Exact and invariant second-moment code format. *J. Engrg. Mech. Div.*, 1974. [192](#), [196](#)
- [61] P. V. Hentenryck. Numerica: a modeling language for global optimization. In *IJCAI'97: Proceedings of the 15th international joint conference on Artificial intelligence*, pages 1642–1647, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. [34](#)
- [62] T. Hickey, Q. Ju, and M. V. Emden. Interval arithmetic: From principles to implementation. *J. ACM*, 48(5):1038–1068, 2001. [18](#)
- [63] M. Hohenbichler and R. Rackwitz. Non-normal dependent vectors in structural safety. *J. Engrg. Mech. Div.*, 107:1227–1238, 1981. [192](#), [196](#), [199](#)
- [64] M. Hohenbichler and R. Rackwitz. First-order concepts in system reliability. *Struct. Safety*, (1):177–188, 1983. [192](#)
- [65] M. Huang, C. Chan, and W. Lou. Optimal performance-based design of wind sensitive tall buildings considering uncertainties. *Comput. Struct.*, 98-99:7–16, 2012. [187](#)
- [66] IEEE. IEEE standard for binary floating-point arithmetic. *ACM SIGPLAN Notices*, 22(2):9–25, 1985. [13](#)

- 
- [67] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer, 2001. [18](#), [31](#), [36](#)
- [68] L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993. [155](#), [158](#)
- [69] W. M. Kahan. A more complete interval arithmetic, 1968. [18](#)
- [70] M. Kalos and P. Whitlock. *Monte Carlo Methods*. John Wiley & Sons, 2009. [73](#)
- [71] R. B. Kearfott. Interval Computations: Introduction, Uses, and Resources. *Euro-math*, Bulletin 2(1):95–112, 1996. [67](#)
- [72] D. Kendall. Foundations of a Theory of Random Sets. In E. Harding and D. Kendall, editors, *Stochastic Geometry*, pages 322–376. John Wiley & Sons, 1974. [45](#)
- [73] A. Kiureghian and T. Dakessian. Multiple design points in 1<sup>st</sup> and 2<sup>nd</sup>-order reliability. *Str. Safety*, 20(1):37–49, 1998. [194](#), [196](#), [200](#), [219](#)
- [74] O. Knüppel. PROFIL/BIAS – A Fast Interval Library. *Computing*, 53:277–287, 1994. [www.ti3.tu-harburg.de/knueppel/profil](http://www.ti3.tu-harburg.de/knueppel/profil). [18](#)
- [75] R. Krawczyk. Newton-algorithmen zur bestimmung von nullstellen mit fehler-schranken. *Computing*, 4(3):187–201, 1969. [26](#)
- [76] V. Kreinovich. Probabilities, intervals, what next? optimization problems related to extension of interval computations to situations with partial information about probabilities. 29(3):265–280, 2004. [80](#)
- [77] N. Kushel and R. Rackwitz. Two basic problems in reliability-based structural optimization. *Math Meth Operat Res*, 46:309–333, 1997. [219](#)
- [78] A. Lakner and R. Anderson. *Reliability engineering for nuclear and other high technology systems : a practical guide*. Elsevier Applied Science Publishers, 1985. [187](#)
- [79] J. Lee, Y. Yang, and W. Ruy. A comparative study on reliability-index and target-performance-based probabilistic structural design optimization. *Computers and Structures*, 80:257–269, 2002. [195](#)

## REFERENCES

---

- [80] Z. Lee, R. Arnone, C. Hu, P. J. Werdell, and B. Lubac. Uncertainties of optical parameters and their propagations in an analytical ocean color inversion algorithm. *Appl. Opt.*, 49(3):369–381, 2010. [175](#), [178](#)
- [81] O. Lhomme. Consistency techniques for numeric CSPs. In *Proc. of the 13th IJCAI*, pages 232–238, 1993. [32](#), [35](#)
- [82] J. Liang, Z. Mourelatos, and J. Tu. A single-loop method for reliability-based design optimisation. *Int. J. Prod. Dev.*, 1-2(5):76–92, 2008. [196](#)
- [83] Q. Lin and J. Rokne. Interval approximation of higher order to the ranges of functions. *Computers and Mathematics with Applications*, 31(7):101 – 109, 1996. [23](#), [24](#)
- [84] K. Makino and M. Berz. Taylor models and other validated functional inclusion methods. *International Journal of Pure and Applied Mathematics*, 4(4):379–456, 2003. [22](#)
- [85] S. Maritorena and D. A. Siegel. Consistent Merging of Satellite Ocean Color Data Sets Using a Bio-optical Model. 94(4):429–440, 2005. [175](#), [178](#)
- [86] C. R. McClain, G. C. Feldman, and S. B. Hooker. An Overview of the SeaWiFS Project and Strategies for Producing a Climate Research Quality Global Ocean Bio-optical Time Series. *Deep Sea Research Part II: Topical Studies in Oceanography*, 51(1-3):5–42, 2004. [179](#)
- [87] R. Melchers. *Structural Reliability Analysis and Prediction*. Civil Engineering Series. John Wiley & Sons, 1999. [192](#), [194](#), [197](#)
- [88] W. Menke. *Geophysical data analysis: discrete inverse theory*. International geophysics series. Academic Press, 1989. [156](#), [157](#)
- [89] U. Montanari. Networks of constraints: Fundamental properties and application to picture processing. *Information Sciences*, 7(2):95–132, 1974. [31](#), [32](#)
- [90] R. Moore. Automatic error analysis in digital computation, 1959. [11](#)
- [91] R. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, 1966. [18](#), [19](#), [20](#), [21](#), [26](#), [34](#), [35](#)

- 
- [92] R. Moore. A test for existence of solutions to nonlinear systems. *SIAM Journal on Numerical Analysis*, 14(4):611–615, 1977. [25](#)
- [93] R. Moore, R. Kearfott, and M. Cloud. *Introduction to Interval Analysis*. SIAM, 2009. [71](#)
- [94] A. Morel and L. Priour. Analysis of Variation in Ocean Colour. *Limnol. Oceanogr.*, 22:709–722, 1977. [176](#)
- [95] A. Nataf. Remarks on a multivariate transformation. *Comptes Rendus de l'Académie des Sciences*, 225(1):42–43, 1962. [192](#)
- [96] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge Univ. Press, 1990. [21](#)
- [97] B. Neveu, G. Trombettoni, and G. Chabert. Improving inter-block backtracking with interval Newton. *Constraints*, 15(1):93–116, 2010. [35](#)
- [98] K. Nickel. On the Newton method in interval analysis, 1971. [25](#)
- [99] E. Nikolaidis, D. Ghiocel, and S. Singhal. *Engineering Design Reliability Applications: For the Aerospace, Automotive and Ship Industries*. CRC Press, 2007. [187](#)
- [100] W. Older and A. Vellino. Constraint arithmetic on real intervals. pages 175–195, 1993. [31](#)
- [101] J. Oxtoby. *Measure and Category*. Springer-Verlag, 1980. [46](#)
- [102] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988. [49](#)
- [103] D. Pestana and S. Velosa. *Introdução à Probabilidade e à Estatística, Volume I*. Fundação Calouste Gulbenkian, Lisboa, 2010. [46](#)
- [104] D. Ratz. On extended interval arithmetic and inclusion monotonicity, 1996. [18](#)
- [105] V. K. Rohatgi. *An introduction to probability theory and mathematical statistics*. Wiley series in probability and mathematical statistics. Wiley, 1976. [56](#), [57](#)

## REFERENCES

---

- [106] M. Rosenblatt. Remarks on a multivariate transformation. *Ann. Math. Stat.*, 3(23):470–472, 1952. [192](#)
- [107] S. M. Rump. INTLAB - INTerval LABoratory. In T. Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999. <http://www.ti3.tu-harburg.de/rump/>. [18](#)
- [108] B. Segovia. *Interactive Light Transport with Virtual Point Lights*. PhD thesis, University Claude Bernard Lyon 1, 2007. [73](#), [74](#)
- [109] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976. [45](#)
- [110] N. Shazeer, M. Littman, and G. Keim. Constraint satisfaction with probabilistic preferences on variable values. In *Proc. of National Conf. on AI*, 1999. [79](#)
- [111] J. Sørensen. Notes in Structural Reliability Theory, 2004. [xvi](#), [195](#), [202](#), [203](#)
- [112] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia, PA, USA, 2004. [155](#), [156](#), [162](#), [168](#)
- [113] G. Trombettoni and G. Chabert. Constructive interval disjunction. In *CP*, pages 635–650, 2007. [35](#)
- [114] Ulrike and Storck. Numerical integration in two dimensions with automatic result verification. In E. Adams and U. Kulisch, editors, *Scientific Computing with Automatic Result Verification*, volume 189 of *Mathematics in Science and Engineering*, pages 187 – 224. Elsevier, 1993. [67](#)
- [115] X. Vu, H. Schichl, and D. Sam-Haroud. Interval propagation and search on directed acyclic graphs for numerical constraint solving. *J. Global Optimization*, 45(4):499–531, 2009. [33](#)
- [116] P. Walley. *Statistical reasoning with imprecise probabilities*. Chapman and Hall, 1991. [45](#)
- [117] T. Walsh. Stochastic constraint programming. In *ECAI*, pages 111–115. IOS Press, 2002. [79](#)

## REFERENCES

---

- [118] P. Wang, E. S. Boss, and C. Roesler. Uncertainties of inherent optical properties obtained from semianalytical inversions of ocean color. *Applied Optics*, 44, 2005. [175](#), [178](#)
- [119] I. Wolfram Research. *Mathematica Edition: Version 8.0*. Wolfram Research, Inc., Champaign, Illinois, 2010. [107](#), [199](#)
- [120] L. Zadeh. Fuzzy sets as a basis for theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978. [45](#)
- [121] G. Zibordi and K. Voss. Field radiometry and ocean colour remote sensing. In V. Barale, J. Gower, and L. Alberotanza, editors, *Oceanography from Space*, chapter 18, pages 307–334. Springer, 2010. [179](#)