Université Pierre et Marie Curie - Paris 6

Mémoire présenté par

Fabienne Jézéquel

en vue d'obtenir l'

Habilitation à diriger des recherches

Dynamical control of approximation methods

Abridged version

soutenue le 10 février 2005 devant le jury composé de :

Götz Alefeld,	Professeur à Universität Karlsruhe, Allemagne,	Examinateur,
Claude Brezinski,	Professeur à l'Université Lille 1,	Rapporteur,
Jean-Marie Chesneaux,	Professeur à l'Université Paris 6,	Examinateur,
Nicholas J. Higham,	Professeur à University of Manchester, Royaume-Uni,	Examinateur,
Bernard Philippe,	Directeur de recherches à l'INRIA, IRISA, Rennes,	Rapporteur,
Marcos Raydan,	Professeur à Universidad Central de Venezuela,	Rapporteur,
Jean Vignes,	Professeur émérite à l'Université Paris 6,	Président.

Abstract

An approximation method provides a result which is affected by a truncation error inherent to the algorithm used and a round-off error due to the finite precision of the computer arithmetic. If the discretization step of such a method decreases, the truncation error also decreases, but the round-off error increases. Therefore it can be difficult to control these two errors simultaneously. Discrete Stochastic Arithmetic (DSA) is an automatic method for round-off error analysis, which uses a random rounding mode. Its computing cost should remain moderate whatever the machine used. DSA has been implemented on vector computer architectures to allow the use of the random rounding mode without inhibiting vectorization. In order to obtain with an approximation method a result for which the global error (consisting of both the truncation error and the round-off error) is minimal, a strategy, based on a converging sequence computation, has been proposed. Computation is carried out until the difference between two successive iterates has no exact significant digit. Then it is possible to determine which digits of the result obtained are in common with the exact solution. This strategy can apply to the computation of integrals using the trapezoidal rule, Simpson's rule, Romberg's method or the Gauss-Legendre method. The different theoretical results which have been established can combine if a numerical method requires the computation of several sequences. If an integral on an infinite domain is approximated by a sum of integrals on finite domains, one can dynamically determine both the optimal number of intervals and the optimal integration step on each interval. A dynamical control of multiple integrals computation has been proposed using two different approaches. The principle of "iterated integrals" consists in expressing a multiple integral as several one-dimensional integrals, which can be evaluated using a quadrature method. Thus one can dynamically determine the optimal integration step for each dimension of the domain considered. A multiple integral can also be evaluated using a cubature method, adapted to the integration domain. The strategy which has been proposed consists in computing a sequence of approximations based on successive partitions of the integration domain. The different theoretical results which have been obtained have been applied to the computation of two improper integrals: one involved in crystallography, the other one in the neutron star theory.

Key-words : round-off error, high performance computing, approximation methods, numerical integration, quadrature methods, cubature methods.

Contents

Introduction

Numerical simulation is used in many fields, such as aironautics, geophysics, climatology or astrophysics, in order to deal with problems for which no simple analytical solution exists. Moreover it enables to avoid experiments which can be expensive (such as crash tests) or even harmful to environment in the case of nuclear tests.
Several types of errors may arise in numerical simulation. First there may be an uncertainty on data measurements. Then in the choice of the equation that describes a physical phenomenon, a modelling error may arise. If an approximation method is used, a truncation error, which depends on the discretization step chosen, is generated. Finally a round-off error is due to the finite precision of the computer arithmetic. It usually arises at each variable assignment and at each arithmetical operation. A computed result may have no more exact significant digit because of round-off error propagation.
In chapter 1, we present a stochastic approach of round-off errors: the CESTAC method and the stochastic arithmetics. Many numerical methods are based on a sequence computation. The strategies described in chapter 2 enable one to dynamically control the number of iterations performed. Thus we obtain an approximation for which the global error, consisting of both the truncation error and the round-off error, is minimal. Then we show how to determine which digits in the result obtained are in common with the exact limit. Chapter 3 is dedicated to the evaluation of multiple integrals. The computing complexity increases with the dimension of the integral studied. The strategies proposed are based on a domain partitioning which can be dynamically controlled. Thus it is possible to identify the digits of the approximation obtained which are in common with the exact value of the integral.

Chapter 1

Stochastic approach of round-off errors

The results of a program run in floating-point arithmetic are usually affected by round-off error propagation. Although other arithmetics exist [37], such as multiprecision arithmetic, exact arithmetics or dynamic arithmetic, their cost is higher, in terms of computing time and memory, and they are sometimes difficult to use. As floating-point arithmetic remains widely spread, several methods have been developed to analyse round-off error propagation, *e.g.* direct analysis [42, 92, 93], inverse analysis [16, 42, 90, 91], methods based on algorithmic differentiation [44, 54, 55, 57, 58, 66, 79], interval arithmetic [4, 48], Monte Carlo arithmetic [70, 71, 72], the CESTAC method [84, 85, 88]. In this chapter, we focus on the stochastic approach of round-off errors. We present the CESTAC method and the principles of stochastic arithmetics.

1.1 The CESTAC method

The CESTAC (Contrôle et Estimation Stochastique des Arrondis de Calculs) method, which has been developed by M. La Porte and J. Vignes [84, 85, 88], enables one to estimate the number of exact significant digits of any computed result. This method is based on a probabilistic approach of round-off errors using a random rounding mode defined below.

Definition 1 *Each real number x, which is not a floating-point number, is bounded by two consecutive floating-point numbers: X^- (rounded down) and X^+ (rounded up). The random rounding mode defines the floating-point number X representing x as being one of the two values X^- or X^+ with the probability $1/2$.*

With this random rounding mode, the same program run several times provides different results, due to different round-off errors.

It has been proved [20] that a computed result R is modelled to the first order in 2^{-k} as:

$$R \approx Z = r + \sum_{i=1}^{n} g_i(d) 2^{-k} z_i \tag{1.1}$$

where r is the exact result, $g_i(d)$ are coefficients depending exclusively on the data and on the code, k is the number of bits in the mantissa and z_i are independent uniformly distributed random variables on $[-1, 1]$.

From equation (1.1), we deduce that:

1. the mean value of the random variable Z is the exact result r,
2. under some assumptions, the distribution of Z is a quasi-Gaussian distribution.

Then by identifying R and Z, *i.e.* by neglecting all the second order terms, Student's test can be used to determine the accuracy of R. Thus from N samples R_i, $i = 1, 2, ..., N$, the number of decimal significant digits common to $\overline{R}$ and r can be estimated with the following equation.

$$C_{\overline{R}} = \log_{10}\left(\frac{\sqrt{N}\left|\overline{R}\right|}{\sigma\tau_\beta}\right), \tag{1.2}$$

where

$$\overline{R} = \frac{1}{N}\sum_{i=1}^{N} R_i \quad \text{and} \quad \sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(R_i - \overline{R}\right)^2. \tag{1.3}$$

τ_β is the value of Student's distribution for $N-1$ degrees of freedom and a probability level $1-\beta$.

Thus the implementation of the CESTAC method in a code providing a result R consists in:

- performing N times this code with the random rounding mode, which is obtained by using randomly the rounding mode towards $-\infty$ or $+\infty$; we then obtain N samples R_i of R
- choosing as the computed result the mean value $\overline{R}$ of R_i, $i = 1, ..., N$
- estimating with equation (1.2) the number of exact decimal significant digits of $\overline{R}$.

In practice $N = 2$ or $N = 3$ and $\beta = 0.05$. Note that for $N = 2$, then $\tau_\beta = 12.706$ and for $N = 3$, then $\tau_\beta = 4.4303$.

Equations (1.1) and (1.2) hold if two main hypotheses are verified. These hypotheses are:

1. the round-off errors α_i are independent, centered uniformly distributed random variables,
2. the approximation to the first order in 2^{-k} is legitimate.

Concerning the first hypothesis, with the use of the random arithmetic, round-off errors α_i are random variables, however, in practice, they are not rigorously centered and in this case Student's test gives a biased estimation of the computed result. It has been proved [27] that, with a bias of a few σ, the error on the estimation of the number of exact significant digits of $\overline{R}$ is less than one decimal digit. Therefore even if the first hypothesis is not rigorously

satisfied, the reliability of the estimation obtained with equation (1.2) is not altered if it is considered as exact up to one digit.
Concerning the second hypothesis, the approximation to the first order only concerns multiplications and divisions. Indeed the round-off error generated by an addition or a subtraction does not contain any term of higher order. It has been shown [20, 24] that, if a computed result becomes insignificant, *i.e.* if the round-off error it contains is of the same order of magnitude as the result itself, then the first order approximation may be not legitimate. In practice the validation of the CESTAC method requires a dynamic control of multiplications and divisions, during the execution of the code. This leads to the synchronous implementation of the method, *i.e.* to the parallel computation of the N samples R_i, and also to the concept of computational zero, also named informatical zero [83].

Definition 2 *During the run of a code using the CESTAC method, an intermediate or a final result R is a computational zero, denoted by @.0, if one of the two following conditions holds:*

- $\forall i, R_i = 0$,
- $C_{\overline{R}} \leq 0$.

Any computed result R is a computational zero if either $R = 0$, R being significant, or R is insignificant. A computational zero is a value that cannot be differentiated from the mathematical zero because of its round-off error.
From the synchronous implementation of the CESTAC method and the concept of computational zero, stochastic arithmetic [24, 28, 85] has been defined. Two types of stochastic arithmetic actually exist: it can be either continuous or discrete.

1.2 Principles of stochastic arithmetics

1.2.1 Continuous stochastic arithmetic

Continuous stochastic arithmetic is a modelling of the synchronous implementation of the CESTAC method. By using this implementation, so that the N runs of a code take place in parallel, the N results of each arithmetical operation can be considered as realizations of a Gaussian random variable centered on the exact result. One can therefore define a new number, called *stochastic number*, and a new arithmetic, called *(continuous) stochastic arithmetic*, applied to these numbers. An equality concept and order relations, which take into account the number of exact significant digits of stochastic operands, have also been defined.
A stochastic number X is denoted by (m, σ^2), where m is the mean value of X and σ its standard deviation. Stochastic arithmetical operations ($s+$, $s-$, $s\times$, $s/$) correspond to terms to the first order in $\frac{\sigma}{m}$ of operations between two independent Gaussian random variables.

Definition 3 *Let $X_1 = (m_1, \sigma_1^2)$ and $X_2 = (m_2, \sigma_2^2)$. Stochastic arithmetical operations on*

X_1 and X_2 are defined as:

$$
\begin{aligned}
X_1 \ s{+} \ X_2 &= \left(m_1 + m_2 \ , \ \sigma_1^2 + \sigma_2^2\right) && (1.4)\\
X_1 \ s{-} \ X_2 &= \left(m_1 - m_2 \ , \ \sigma_1^2 + \sigma_2^2\right) && (1.5)\\
X_1 \ s{\times} \ X_2 &= \left(m_1 \times m_2 \ , \ m_2^2\sigma_1^2 + m_1^2\sigma_2^2\right) && (1.6)\\
X_1 \ s/ \ X_2 &= \left(m_1/m_2 \ , \ \left(\frac{\sigma_1}{m_2}\right)^2 + \left(\frac{m_1\sigma_2}{m_2^2}\right)^2\right) \textit{with} \ \ m_2 \neq 0. && (1.7)
\end{aligned}
$$

An accuracy can be associated to any stochastic number. If $X = (m, \sigma^2)$, λ_β exists (depending only on β) such that

$$P\left(X \in [m - \lambda_\beta\sigma, m + \lambda_\beta\sigma]\right) = 1 - \beta, \tag{1.8}$$

$I_{\beta,X} = [m - \lambda_\beta\sigma, m + \lambda_\beta\sigma]$ is the confidence interval of m at $1 - \beta$. The number of decimal significant digits common to all the elements of $I_{\beta,X}$ and to m is lower bounded by

$$C_{\beta,X} = \log_{10}\left(\frac{|m|}{\lambda_\beta\sigma}\right). \tag{1.9}$$

The following definition is the modelling of the concept of computational zero, previously introduced.

Definition 4 *A stochastic number X is a stochastic zero, denoted by $\underline{0}$, if and only if*

$$C_{\beta,X} \leq 0 \ \ \textit{or} \ \ X = (0, 0).$$

In accordance with the concept of stochastic zero, a new equality concept and new order relations have been defined.

Definition 5 *Let $X_1 = (m_1, \sigma_1^2)$ and $X_2 = (m_2, \sigma_2^2)$ be two stochastic numbers.*

- *Stochastic equality, denoted by $s{=}$, is defined as:*
  $X_1 \ s{=} \ X_2$ *if and only if* $X_1 \ s{-} \ X_2 = \underline{0}$.

- *Stochastic inequalities, denoted by $s{>}$ and $s{\geq}$ are defined as:*
  $X_1 \ s{>} \ X_2$ *if and only if* $m_1 > m_2$ *and* $X_1 \ s{\neq} \ X_2$,
  $X_1 \ s{\geq} \ X_2$ *if and only if* $m_1 \geq m_2$ *or* $X_1 \ s{=} \ X_2$.

Continuous stochastic arithmetic is a modelling of the computer arithmetic, which takes into account round-off errors. The properties of continuous stochastic arithmetic [23, 24] have pointed out the theoretical differences between the approximative arithmetic of a computer and exact arithmetic.

1.2.2 Discrete Stochastic Arithmetic

Discrete Stochastic Arithmetic (DSA) has been defined from the synchronous implementation of the CESTAC method. With DSA, a real number becomes an N-dimensional set and any operation on these N-dimensional sets is performed element per element using the random rounding mode. The number of exact significant digits of such an N-dimensional set can be estimated from equation (1.2). From the concept of computational zero previously introduced, an equality concept and order relations have been defined for DSA.

Definition 6 *Let X and Y be N-samples provided by the CESTAC method.*

- *Discrete stochastic equality denoted by $ds{=}$ is defined as:*
  $X ds{=} Y$ *if and only if* $X - Y = @.0$.
- *Discrete stochastic inequalities denoted by $ds{>}$ and $ds{\geq}$ are defined as:*
  $X ds{>} Y$ *if and only if* $\overline{X} > \overline{Y}$ *and* $X ds{\neq} Y$,
  $X ds{\geq} Y$ *if and only if* $\overline{X} \geq \overline{Y}$ *or* $X ds{=} Y$.

Order relations in DSA are essential to control branching statements. Because of round-off errors, if A and B are two floating-point numbers and a and b the corresponding exact values,

$$a > b \not\Rightarrow A > B \quad \text{and} \quad A > B \not\Rightarrow a > b.$$

Many problems in scientific computing are due to this dis-correlation: for example, unsatisfied stopping criteria or infinite loops in algorithmic geometry. Taking into account the numerical quality of the operands in order relations enables to partially solve these problems [23].
Therefore DSA enables to estimate the impact of round-off errors on any result of a scientific code and also to check that no anomaly occurred during the run, especially in branching statements. DSA is implemented in the CADNA (Control of Accuracy and Debugging for Numerical Applications) library [14, 24].

The accuracy of a stochastic number can be related to the number of exact significant digits of an N-sample provided by the CESTAC method. Indeed, when N is a small value (2 or 3), which is the case in practice, the values obtained with equations (1.2) and (1.9) are very close. They represent in a computed result the number of significant digits which are not affected by round-off errors. So the two types of stochastic arithmetics are coherent. Properties established in the theoretical framework of continuous stochastic arithmetic can be applied on a computer via the practical use of DSA.

1.3 Control of stochastic operations

The theoretical results presented here have been established in continuous stochastic arithmetic. They enable one to compare results of arithmetical stochastic operations with those provided by the corresponding classical operations performed on exact values.
Let us consider a numerical method which aims to approximate an exact value x_1. This method may consist for example in computing an iterate of a sequence (u_n) such that

$\lim_{n\to\infty} u_n = x_1$. Even using an arithmetic with infinite precision, the value obtained is not x_1, but an approximation which is affected by a truncation error. We compare here the results obtained using such numerical methods in stochastic arithmetic with the exact values they approximate.

Theorem 1 *Let $X_1 = (m_1, \sigma_1^2)$ be the approximation of an exact value x_1 in stochastic arithmetic. Let us assume that the exact significant bits of X_1, i.e. not affected by round-off errors, are in common with x_1, up to p: the number of significant bits of X_1 in common with x_1 is lower bounded by $log_2\left(\frac{|m_1|}{\lambda_\beta \sigma_1}\right) - p$.*
Similarly let $X_2 = (m_2, \sigma_2^2)$ be an approximation obtained in stochastic arithmetic of an exact value x_2, such that its exact significant bits are in common with x_2, up to q.
Let $\bigcirc$ be an exact arithmetical operator: $\bigcirc \in \{+, -, \times, /\}$ and $s\bigcirc$ the corresponding stochastic operator $s\bigcirc \in \{s+\,, s-\,, s\times\,, s/\}$.
Then the exact significant bits of $X_1\ s\bigcirc\ X_2$ are in common with the exact value $x_1 \bigcirc x_2$, up to $max(p, q)$.

The proof, established in [45], can be consulted in 5.3.

Theorem 1 enables one to control arithmetical operations performed on computed results of numerical methods. This theorem has been proved for stochastic arithmetical operations, which are a modelling of the operations performed in the synchronous implementation of the CESTAC method. In practice, theorem 1 is used, according to 1.2.2, for results obtained in DSA. In the next section, we present, in accordance with theorem 1 and the theoretical results presented in section 2.1, a strategy to dynamically control converging sequences computed in DSA.

1.4 Implementation of Discrete Stochastic Arithmetic

DSA is implemented in the CADNA library [14, 24], which can be used to control the numerical quality of any scientific code written in ADA, C or Fortran. The CADNA library contains the definition of new numerical types: the stochastic types. All arithmetical operators and order relations have been overloaded in accordance with the properties of DSA. Similarly the input/output statements have been modified, especially the printing statement, which gives the result written only with its exact decimal significant digits.
The use of the CADNA library requires only a few modifications in the initial code as: replacing the declaration statements of classical floating-point types by stochastic types and changing some statements such as printing statements. During the run, as soon as a numerical anomaly (appearance of a computational zero) occurs, a message is written in a special file. The CADNA library is also able to estimate the impact of data errors on the results provided by a code.
J.-M. Chesneaux wrote the first version of the CADNA library in ADA [18, 19, 21]. He developed later a version in Fortran 90 [22] which could initially control the numerical quality of programs written in Fortran 77. Then a version which can take into account array notations of Fortran 90 has been developed on the occasion of the implementation of CADNA

on vector computers [46]. Besides, a version in C++ has been written by J.-L. Lamotte. A version in Fortran and a version in C++ can be downloaded from the web site dedicated to the CADNA library[1].

In section 5.2, we describe the implementation of CADNA on two vector computers: CRAY SV1 and NEC SX5. Two types of versions of the CADNA library have been developed for parallel architectures. One can distinguish the versions which enable to control the numerical quality of parallel codes [8, 49], from those which take advantage of the different processors of a parallel machine to improve the performances of CADNA for sequential codes [7, 8, 50, 51].

1.5 Conclusion

During these last years, both theoretical and practical studies on stochastic arithmetic have been carried out.

Theorem 1, given in section 1.3, enables one to determine which significant digits in the computed result of an arithmetical operation are in common with the exact result. Established in continuous stochastic arithmetic, it is used in practice for results computed in DSA.
From the concept of computational zero, described in section 1.1, an optimal stopping criterion for iterative algorithms [5, 84, 85, 86, 87] has been proposed. In the next chapters, we show that the combination of the optimal stopping criterion with new theoretical results, such as theorem 1, enables one to take into account both the round-off error and the truncation error in several numerical methods.
By studying addition and multiplication by scalars in continuous stochastic arithmetic, S. Markov and R. Alt have recently shown new properties of stochastic numbers [65].

Several practical studies of DSA have been carried out in order to improve its performances and to adapt it to the floating-point arithmetic of the processor, to the architecture or to the compiler of the machine used. Furthermore DSA has been implemented for fixed-point arithmetic [25]. Besides the improvements of the CADNA library, a specific hardware dedicated to DSA has been developed by R. Chotin-Avot during her PhD thesis [29].

[1] `http://www.lip6.fr/cadna`

Chapter 2

Dynamical control of converging sequences computation

Many numerical methods are based on sequences computation: Newton's method to evaluate roots of functions, Runge-Kutta formulas to solve differential equations, finite difference schemes to solve partial differential equations, Krylov methods to solve linear systems...
If a discretization step, which is related to the number of iterations, decreases, then the truncation error decreases, but the round-off error increases. An estimation of the truncation error is usually performed in scientific libraries in order to respect a threshold chosen by the user.
We propose to take into account the global error, consisting of both the truncation error and the round-off error, which is generated by converging sequences computation. The strategies described in this chapter enable one to dynamically compute the iterate which minimizes the global error and to determine its significant digits in common with the exact limit. These strategies can apply to the computation of integrals. Indeed a sequence of approximations can be generated by using a composite quadrature method, *i.e.* based on successive partitions of the integration domain. Thus one can dynamically control the computation of integrals with the trapezodial rule, Simpson's rule, Romberg's method or the Gauss-Legendre method. Furthermore we describe a strategy for numerical methods combining several converging sequences, which can be applied to the computation of integrals on an infinite domain.

2.1 Theoretical results on converging sequences

2.1.1 Preliminary definitions

The theorems presented here have been established for sequences having a linear or an exponential convergence speed. Therefore we recall properties which characterize these two types of convergence speed.

Definition 7 *A sequence (I_n) converges to I with a linear speed if*

$$I_n - I = K\alpha^n + o(\alpha^n), \text{ where } K \in \mathbb{R} \text{ and } 0 < |\alpha| < 1.$$

With a sequence having a linear convergence, the number of iterations required to obtain an approximation of the limit with one more exact digit is quasi-constant.

Definition 8 *A sequence (I_n) converges to I with an exponential speed if*

$$I_n - I = K\,\alpha^{p^n} + o(\alpha^{p^n}), \text{ where } K \in \mathbb{R},\ 0 < |\alpha| < 1 \text{ and } p > 1.$$

With a sequence having an exponential convergence, at each iteration, the number of exact digits is quasi-multiplied by p.
The theoretical results presented in this section require the notion of significant digits common to two real numbers. Therefore we need the following definition.

Definition 9 *Let a and b be two real numbers, the number of significant digits that are common to a and b can be defined in $\mathbb{R}$ by*

1. *for $a \neq b$, $C_{a,b} = \log_{10}\left|\frac{a+b}{2(a-b)}\right|$,*
2. *$\forall a \in \mathbb{R},\ C_{a,a} = +\infty$.*

Then $|a-b| = \left|\frac{a+b}{2}\right| 10^{-C_{a,b}}$. For instance, if $C_{a,b} = 3$, the relative difference between a et b is of the order of 10^{-3} which means that a and b have three significant digits in common.

Remark 1 *The value of $C_{a,b}$ can seem surprising if we consider the decimal notations of a and b. For example, if $a = 2.4599976$ and $b = 2.4600012$, then $C_{a,b} \approx 5.8$. The difference due to the sequences of "0" or "9" is illusive. The significant decimal digits of a and b are really different from the sixth position.*

2.1.2 On sequences with a linear convergence

Let us consider a sequence (I_n) converging linearly to I. From the number of significant digits common to two successive iterates, I_n and I_{n+1}, the following theorem enables one to determine the number of significant digits common to I_n and the exact limit I.

Theorem 2 *Let (I_n) be a sequence converging linearly to I, i.e. which satisfies $I_n - I = K\alpha^n + o(\alpha^n)$ where $K \in \mathbb{R}$ and $0 < |\alpha| < 1$, then*

$$C_{I_n,I_{n+1}} = C_{I_n,I} + \log_{10}\left(\frac{1}{1-\alpha}\right) + o(1).$$

The proof, established in [45], can be consulted in 5.3.

If the convergence zone is reached, *i.e.* if the term $o(1)$ becomes negligible, the significant decimal digits common to I_n and I_{n+1}, are also in common with the exact limit I, up to $\log_{10}\left(\frac{1}{1-\alpha}\right)$.

If $\alpha \in (-1,0)$, then $-\log_{10} 2 < \log_{10}\left(\frac{1}{1-\alpha}\right) < 0$. In this case, if the convergence zone is reached, the significant digits in common between I_n and I_{n+1} are also in common with I.
$\forall \alpha \in (0,1)$, $\exists k\ 0 < \alpha \leq 1 - 10^{-k}$ and therefore $0 < \log_{10}\left(\frac{1}{1-\alpha}\right) \leq k$. If the convergence zone is reached, the significant digits in common between I_n and I_{n+1} are also in common with I, up to k digits. The lower α is, the faster the convergence of the sequence is and the lower k is.

Remark 2 *If* $0 < \alpha \le \frac{1}{2}$*, then* $0 < \log_2\left(\frac{1}{1-\alpha}\right) \le 1$*. In this case, if the convergence zone is reached, the significant bits in common between* I_n *and* I_{n+1} *are also in common with* I*, up to one.*

2.1.3 On sequences with an exponential convergence

Theoretical results similar to theorem 2 may be established for sequences with an exponential convergence.

Theorem 3 *Let* (I_n) *be a sequence converging to* I *with an exponential speed, i.e. which satisfies* $I_n - I = K\,\alpha^{p^n} + o(\alpha^{p^n})$ *where* $K \in I\!R$*,* $0 < |\alpha| < 1$ *and* $p > 1$*, then*

$$C_{I_n,I_{n+1}} = C_{I_n,I} + \log_{10}\left(\frac{1}{1-\alpha^{p^n(p-1)}}\right) + o\left(1\right).$$

The proof, established in [45], can be consulted in 5.3.

If the convergence zone is reached, the decimal significant digits in common between I_n and I_{n+1} are also common to the exact limit I, up to $\log_{10}\left(\frac{1}{1-\alpha^{p^n(p-1)}}\right)$.
If $0 < |\alpha| \le M_n$, with $M_n = (\frac{9}{10})^{\left(\frac{1}{p^n(p-1)}\right)}$, then $0 < \log_{10}\left(\frac{1}{1-\alpha^{p^n(p-1)}}\right) \le 1$. The significant digits common to I_n and I_{n+1} are also common to I, up to one. As the number n of iterations increases, M_n also increases and the condition that α must satisfy in order to have $\log_{10}\left(\frac{1}{1-\alpha^{p^n(p-1)}}\right) \le 1$ becomes less and less strict. For example, if the sequence (I_n) has a quadratic convergence, which is characterized by $p = 2$, then $M_1 > 0.94$ and $M_5 > 0.99$. Similarly, as p increases, the speed of convergence increases and M_n also increases.

Remark 3 *If the convergence zone is reached, the significant bits in common between* I_n *and* I_{n+1} *are also common to the exact limit* I*, up to* $\log_2\left(\frac{1}{1-\alpha^{p^n(p-1)}}\right)$*. If* $0 < |\alpha| \le 2^{\left(\frac{1}{p^n(1-p)}\right)}$*, then* $0 < \log_2\left(\frac{1}{1-\alpha^{p^n(p-1)}}\right) \le 1$*. This condition on* α *is easily satisfied. Indeed in the case of a quadratic convergence (i.e. for* $p = 2$*) if* $n = 5$*,* $2^{\left(\frac{1}{p^n(1-p)}\right)} > 0.97$*.*

2.1.4 Applications in Discrete Stochastic Arithmetic

When a numerical algorithm requires the evaluation of the limit of a sequence, this limit is approximated by one of the iterates. As the number of iterations increases, the truncation error usually decreases, but the round-off error increases. Therefore the choice of the optimal iterate may be problematic.
DSA enables one to estimate the number of exact significant digits of any computed result, *i.e.* its significant digits which are not affected by round-off error propagation. Let us consider the computation of a sequence (I_n) in DSA and let us assume that the convergence zone is reached. If discrete stochastic equality is achieved for two successive iterates, *i.e.* $I_n - I_{n+1} = @.0$, the difference between I_n and I_{n+1} is only due to round-off errors and further iterations are useless. The optimal iterate I_{n+1} can therefore be dynamically determined at run time. Furthermore, if the sequence (I_n) converges at least linearly to I, from 2.1.2 and

2.1.3, the exact significant digits of I_{n+1} are in common with I, up to k digits. The value k, which depends on the convergence speed of (I_n), can be determined from theorem 2 or 3.

2.2 Dynamical control of approximation methods

A numerical method which uses a discretization step h enables one to approximate an exact value L by a value $L(h)$ such that $\lim_{h\to 0} L(h) = L$. If the step h decreases, the truncation error on $L(h)$ also decreases, but the round-off error usually increases. Therefore it can be difficult to determine the step value which minimizes the global error, consisting of both the truncation error and the round-off error. We propose a strategy, based on several successive divisions of the step by 2, which enables one to dynamically determine the optimal step value. Using theorem 4, we show that the exact significant bits of the approximation obtained are in common with the exact result, up to one.

Theorem 4 *Let us consider a numerical method which provides an approximation $L(h)$ of order p of an exact value L, i.e. $L(h) - L = Kh^p + \mathcal{O}\left(h^q\right)$ with $1 \le p < q,\ K \in \mathbb{R}$.*
If L_n is the approximation computed with the step $\frac{h_0}{2^n}$, then

$$C_{L_n,L_{n+1}} = C_{L_n,L} + log_{10}\left(\frac{2^p}{2^p-1}\right) + \mathcal{O}\left(2^{n(p-q)}\right).$$

Proof The truncation error on L_n is

$$L_n - L = K\left(\frac{h_0}{2^n}\right)^p + \mathcal{O}\left(\frac{1}{2^{qn}}\right) \tag{2.1}$$

Using the same formula for L_{n+1}, one obtains

$$L_n - L_{n+1} = K\ \left(\frac{2^p-1}{2^p}\right)\ \left(\frac{h_0}{2^n}\right)^p + \mathcal{O}\left(\frac{1}{2^{qn}}\right) \tag{2.2}$$

From equation (2.1), we deduce

$$\frac{L_n}{L_n - L} = \frac{L_n}{K(\frac{h_0}{2^n})^p\,(1+\mathcal{O}(2^{n(p-q)}))} \tag{2.3}$$

$$\frac{L_n}{L_n - L} = \frac{L_n}{K(\frac{h_0}{2^n})^p}\left(1+\mathcal{O}(2^{n(p-q)})\right) \tag{2.4}$$

Therefore

$$\frac{L_n}{L_n - L} = \frac{L_n}{K(\frac{h_0}{2^n})^p} + \mathcal{O}(2^{n(2p-q)}) \tag{2.5}$$

Then

$$\frac{L_n + L}{2(L_n - L)} = \frac{L_n}{L_n - L} - \frac{1}{2} = \frac{L_n}{K(\frac{h_0}{2^n})^p} + \mathcal{O}(2^{n(2p-q)}) \tag{2.6}$$

Similarly, from equation (2.2), we deduce

$$\frac{L_n + L_{n+1}}{2(L_n - L_{n+1})} = \frac{L_n}{L_n - L_{n+1}} - \frac{1}{2} = \left(\frac{L_n}{K(\frac{h_0}{2^n})^p}\right)\left(\frac{2^p}{2^p-1}\right) + \mathcal{O}(2^{n(2p-q)}) \tag{2.7}$$

From definition 9 and equation (2.6), we deduce

$$C_{L_n,L} = \log_{10}\left|\frac{L_n}{K(\frac{h_0}{2^n})^p}\left(1+\mathcal{O}(2^{n(p-q)})\right)\right| \tag{2.8}$$

$$C_{L_n,L} = \log_{10}\left|\frac{L_n}{K(\frac{h_0}{2^n})^p}\right| + \log_{10}\left|1+\mathcal{O}(2^{n(p-q)})\right| \tag{2.9}$$

Therefore

$$C_{L_n,L} = \log_{10}\left|\frac{L_n}{K(\frac{h_0}{2^n})^p}\right| + \mathcal{O}\left(2^{n(p-q)}\right) \tag{2.10}$$

Similarly, from definition 9 and equation (2.7), we deduce

$$C_{L_n,L_{n+1}} = \log_{10}\left|\left(\frac{L_n}{K(\frac{h_0}{2^n})^p}\right)\left(\frac{2^p}{2^p-1}\right)\right| + \mathcal{O}\left(2^{n(p-q)}\right) \tag{2.11}$$

Finally

$$C_{L_n,L_{n+1}} = C_{L_n,L} + \log_{10}\left(\frac{2^p}{2^p-1}\right) + \mathcal{O}\left(2^{n(p-q)}\right). \tag{2.12}$$

□

If the convergence zone is reached, *i.e.* if the term $\mathcal{O}\left(2^{n(p-q)}\right)$ becomes negligible, the significant digits common to two successive approximations L_n and L_{n+1} are also in common with the exact result L, up to one bit. Indeed the term $\log_{10}\left(\frac{2^p}{2^p-1}\right)$ decreases as p increases and it corresponds to one bit for methods of order 1.
One can dynamically determine the optimal approximation by performing computations until the difference $L_n - L_{n+1}$ has no exact significant digit. If the convergence zone has been reached, then the exact significant bits of the last approximation are in common with L, up to one.

2.3 Dynamical control of quadrature methods

2.3.1 The trapezoidal rule and Simpson's rule

Like for other approximation methods, the dynamical control of the trapezoidal rule or Simpson's rule is based on successive divisions of the integration step by 2. A sequence of approximations, which converges linearly to the integral, is generated. Theorems 5 and 6 enable one to determine, from two successive approximations, the first digits of the exact value of the integral.

Theorem 5 *Let us assume that f is a real function which is C^k over $[a,b]$ where $k \geq 3$ and that $f'(b) \neq f'(a)$. Let I_n be the approximation of $I = \int_a^b f(x)dx$ by the trapezoidal rule with step $h = \frac{b-a}{2^n}$. Then*

$$C_{I_n,I_{n+1}} = C_{I_n,I} + log_{10}\left(\frac{4}{3}\right) + \mathcal{O}\left(\frac{1}{4^n}\right). \tag{2.13}$$

Theorem 6 *Let us assume that f is a real function which is C^k over $[a,b]$ where $k \geq 5$ and that $f^{(3)}(b) \neq f^{(3)}(a)$. Let I_n be the approximation of $I = \int_a^b f(x)dx$ by Simpson's rule with step $h = \frac{b-a}{2^n}$. Then*

$$C_{I_n,I_{n+1}} = C_{I_n,I} + log_{10}\left(\frac{16}{15}\right) + \mathcal{O}\left(\frac{1}{4^n}\right). \tag{2.14}$$

The proofs of these two theorems, established in [26], can be consulted in 5.1.

If the convergence zone is reached, *i.e.* if the term $\mathcal{O}\left(\frac{1}{4^n}\right)$ becomes negligible, the significant digits common to two successive approximations I_n and I_{n+1} are also in common with I, the exact value of the integral, up to one. Indeed the terms $log_{10}\left(\frac{4}{3}\right)$ and $log_{10}\left(\frac{16}{15}\right)$ correspond at most to one bit.

If computations are performed until, in the convergence zone, the difference between two successive approximations has no exact significant digit, then the significant bits of the last approximation which are not affected by round-off errors are in common with the exact value of the integral, up to one.

Theorems 5 and 6 have been established from the truncation error expansion on I_n [77],

- for the trapezoidal rule:

$$I_n - I = \frac{h^2}{12}\left[f'(b) - f'(a)\right] + \mathcal{O}(h^4) \tag{2.15}$$

- and for Simpson's rule:

$$I_n - I = \frac{h^4}{180}\left[f^{(3)}(b) - f^{(3)}(a)\right] + \mathcal{O}(h^6) \tag{2.16}$$

with $h = \dfrac{b-a}{2^n}$.

These equations are similar to equation (2.1) which characterizes approximation methods, with $p = 2$ and $q = 4$ for the trapezoidal rule; $p = 4$ and $q = 6$ for Simpson's rule. This explains the coherence of theorems 5 and 6 with theorem 4.

Numerical results obtained by performing a dynamical control of computations using the trapezoidal rule or Simpson's rule are described in sections 5.1 and 5.3.

2.3.2 The Gauss-Legendre method

The approximation of $\int_{-1}^{1} f(x)dx$ by the Gauss-Legendre method with ν points [30, 39] is $\sum_{i=1}^{\nu} C_i f(x_i)$, where for $i = 1, ..., \nu$, $\{x_i\}$ are the roots of the ν-degree Legendre polynomial P_ν and

$$C_i = \frac{2}{(1 - x_i^2)(P'_\nu(x_i))^2}. \tag{2.17}$$

For the computation of an integral on another interval such as $I = \int_a^b g(t)dt$, the following change of variable is required.

$$I = \int_a^b g(t)dt = \frac{b-a}{2} \int_{-1}^{1} f(x)dx, \tag{2.18}$$

with

$$\forall x \in [-1, 1],\ f(x) = g\left(\frac{(b-a)x + b + a}{2}\right). \tag{2.19}$$

The Gauss-Legendre method with ν points is of order 2ν: it is exact if the integrand is a polynomial of degree r with $r \leq 2\nu - 1$.

We propose a strategy for a dynamical control of the Gauss-Legendre method although this method uses no integration step. This strategy is based on partitions of the integration interval into subintervals of same length on which the classical Gauss-Legendre method is applied.
Let us assume that the integration interval is partitioned into 2^n subintervals and that the integral on each subinterval is evaluated using the Gauss-Legendre method with ν points. Theorem 7 presents the truncation error on I_n, the sum of the 2^n approximations obtained.

Theorem 7 *Let g be a C^p function on $[a, b]$ with $p \geq 2\nu$.*
Let $I = \int_a^b g(t)dt$ and for $i = 1, ..., \nu$, let $\{x_i\}$ be the roots of the ν-degree Legendre polynomial and $\{C_i\}$ the corresponding weights.
We assume that the integral on each subinterval $[a + (k-1)\frac{b-a}{2^n},\ a + k\frac{b-a}{2^n}]$ ($k = 1, ..., 2^n$) is evaluated using the Gauss-Legendre method with ν points.
Let I_n be the sum of the 2^n approximations obtained.
Then

$$I_n - I = \frac{K_\nu}{4^{n\nu}} + \mathcal{O}\left(\frac{1}{2^{n(2\nu+1)}}\right)$$

with $K_\nu = \frac{(b-a)^{2\nu}}{2^{2\nu+1}(2\nu)!}\left(\sum_{i=1}^{\nu} C_i x_i^{2\nu} - \frac{2}{2\nu+1}\right)\left[g^{(2\nu-1)}(b) - g^{(2\nu-1)}(a)\right].$

In section 5.5, one can consult a more general form of theorem 7, where the integration interval is partitioned into q subintervals.

Theorem 8 enables one, from the significant digits common to two successive approximations, I_n and I_{n+1}, to determine the significant digits common to I_n and I, the exact value of the integral. The same notations and assumptions as in theorem 7 are used.

Theorem 8

$$C_{I_n,I_{n+1}} = C_{I_n,I} + log_{10}\left(\frac{4^\nu}{4^\nu - 1}\right) + \mathcal{O}\left(\frac{1}{2^n}\right).$$

In section 5.5, one can consult a more general form of theorem 8, where an iterate is obtained by q partitions of the integration interval and the next iterate by $q+s$ partitions. Thus theorem 8 corresponds to $q = s = 2^n$.

If the convergence zone is reached, *i.e.* if the term $\mathcal{O}\left(\frac{1}{2^n}\right)$ becomes negligible, the significant digits common to two successive approximations are also common to exact value of the integral, up to one bit. Indeed, for any ν, the term $log_{10}\left(\frac{4^\nu}{4^\nu-1}\right)$ corresponds at most to one bit.

If computations are performed until, in the convergence zone, the difference between two successive approximations has no exact significant digit, then the significant bits of the last approximation which are not effected by round-off errors are in common with the exact value of the integral, up to one.

The truncation error established in theorem 7 is coherent with equation (2.1) which characterizes approximation methods, with $p = 2\nu$, $q = 2\nu+1$ and $h_0 = b-a$. Therefore theorem 8 could have been established from theorem 4, with $p = 2\nu$ and $q = 2\nu + 1$.

In section 5.5, we describe the dynamical control of the computation of an improper integral with the Gauss-Legendre method.

2.3.3 Romberg's method

Romberg's method [10, 13, 77] is based on Richardson's extrapolation on results of the trapezoidal rule. For the approximation of $I = \int_a^b f(x)dx$, Romberg's method consists in computing the following triangular table, with $h = \frac{b-a}{M}$ $(M \geq 1)$.

$$\begin{array}{llllll}
T_1(h) & T_1(\frac{h}{2}) & .... & T_1(\frac{h}{2^{n-3}}) & T_1(\frac{h}{2^{n-2}}) & T_1(\frac{h}{2^{n-1}}) \\
T_2(h) & T_2(\frac{h}{2}) & .... & T_2(\frac{h}{2^{n-3}}) & T_2(\frac{h}{2^{n-2}}) & \\
T_3(h) & T_3(\frac{h}{2}) & .... & T_3(\frac{h}{2^{n-3}}) & & \\
\vdots & \vdots & & & & \\
T_{n-1}(h) & T_{n-1}(\frac{h}{2}) & & & & \\
T_n(h) & & & & &
\end{array}$$

The first row of the table represents approximations of I computed using the trapezoidal rule with step $\frac{h}{2^j}$ $(j = 0, ..., n-1)$. Rows 2 to n are computed using the following formula.

For $r = 2, ..., n$ and $j = 0, ..., n-r$,

$$T_r\left(\frac{h}{2^j}\right) = \frac{1}{4^{r-1}-1}\left(4^{r-1}T_{r-1}\left(\frac{h}{2^{j+1}}\right) - T_{r-1}\left(\frac{h}{2^j}\right)\right). \tag{2.20}$$

The sequence of approximations with Romberg's method $T_1(h), ..., T_n(h)$ converges exponentially to I.
The following theorem enables one to determine from two successive approximations, $T_n(h)$ and $T_{n+1}(h)$, the first digits of the exact value of the integral.

Theorem 9 *We assume that f is a real function which is $\mathcal{C}^k$ over $[a,b]$ where $k \geq 2n+1$ and that $f^{(2n-1)}(a) \neq f^{(2n-1)}(b)$. Let $T_n(h)$ be the approximation of $I = \int_a^b f(x)dx$ computed with n iterations of Romberg's method using the initial step $h = \frac{b-a}{M}$ $(M \geq 1)$. Then*

$$C_{T_n(h),T_{n+1}(h)} = C_{T_n(h),I} + \mathcal{O}\left(\frac{1}{n^2}\right).$$

The proof of this theorem, established in [47], can be consulted in section 5.4.

If the convergence zone is reached, *i.e.* if $O\left(1/n^2\right) \ll 1$, the significant digits common to two successive approximations, $T_n(h)$ and $T_{n+1}(h)$, are also in common with the exact value of the integral.

If computations are performed until, in the convergence zone, the difference between two successive approximations has no exact significant digit, then the significant digits of the last approximation which are not affected by round-off errors are in common with the exact value of the integral.

The equation given in theorem 9 is different from the one in theorem 4, established for approximation methods. Indeed Romberg's method is not an approximation method with an order known beforehand. The approximation $T_n(h)$ obtained with n iterations of Romberg's method is exact if the integrand is a polynomial of degree r with $r \leq 2n-1$. Moreover the dynamical control of Romberg's method is not based on successive divisions of an integration step. This dynamical control consists in comparing two successive approximations, $T_n(h)$ and $T_{n+1}(h)$, and not $T_n(h)$ and $T_n\left(\frac{h}{2}\right)$.

In section 5.4, we describe the dynamical control of the computation of an improper integral with Romberg's method.

2.4 Dynamical control of combined sequences

In this section, we show how to approximate the limit of a sequence by its optimal iterate, this iterate being itself the limit of another sequence. The theorems established in chapters 1 and 2 can be combined to determine the digits of the approximation obtained which are in common with the exact result. In the strategies described in this section, small letters denote exact values and capital letters the corresponding approximations computed using DSA.

2.4.1 A strategy to compute combined sequences

We consider a sequence in which each term u_m is the limit of another sequence. More precisely, let (u_m) be a sequence converging at least linearly to u and, for all m, let $(u_{m,n})$ be a sequence converging at least linearly to u_m.
For all m, let U_m be the approximation of u_m computed using DSA. U_m is obtained by computing the sequence $(u_{m,n})$ until, in the convergence zone, the difference between two successive iterates has no exact significant digit.

As for all m, the sequence $(u_{m,n})$ converges at least linearly to u_m, according to section 2.1, one can determine the value q such that the exact significant bits of U_m are common to u_m, up to q.
Figure 2.1 presents the significant bits of U_m and U_{m+1} if the difference $U_m - U_{m+1}$ has no exact significant digit. In this case, the exact significant bits of U_{m+1} are common to U_m and are also common to u_m and u_{m+1}, up to q.

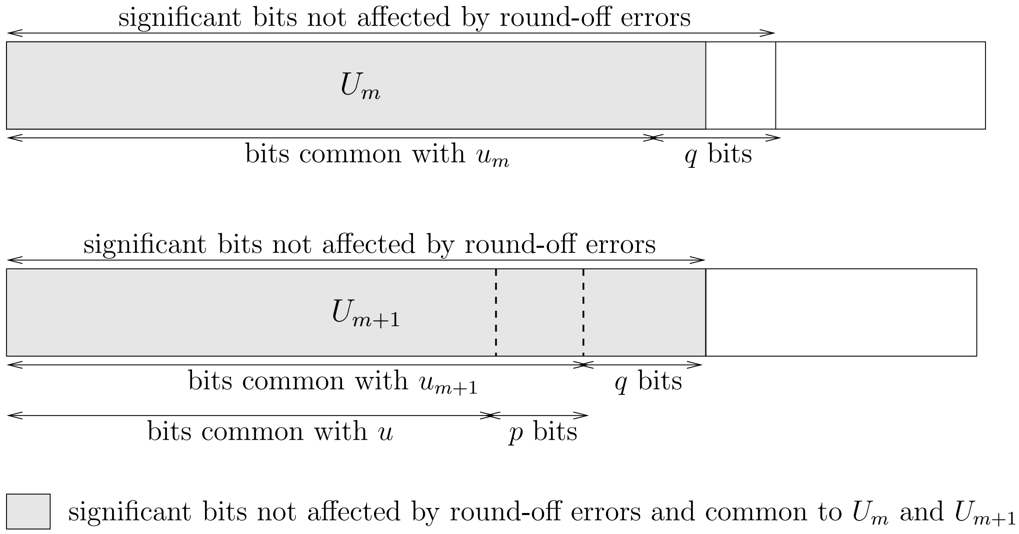

Figure 2.1: Significant bits of U_m and U_{m+1}

As the sequence (u_m) converges at least linearly to u, one can determine the value p such that the bits common to u_m and u_{m+1} are common with u, up to p.
Consequently if the difference $U_m - U_{m+1}$ has no exact significant digit, the exact significant bits of U_{m+1} are common with u, up to $p + q$.

2.4.2 Dynamical control of integrals on an infinite domain

Let us consider the computation of an improper integral $g = \int_0^\infty \phi(x)dx$. The infinite interval of integration is partitioned into finite intervals of length L. Let $f_j = \int_{jL}^{(j+1)L} \phi(x)dx$ and $g_m = \sum_{j=0}^{m} f_j$, $\lim_{m\to\infty} g_m = g$.

The exact value g can be numerically approximated by an iterate g_m, m being sufficiently high. The optimal number of iterates to compute can be determined dynamically using DSA.

Each integral f_j can be evaluated using one of the strategies described in section 2.3. Let $F_{j,n}$ be the approximation of f_j computed:

- by using the trapezoidal rule or Simpson's rule with step $\frac{L}{2^n}$,
- by partitioning the integration interval into 2^n subintervals, on which the Gauss-Legendre method is used
- or by performing n iterations of Romberg's method.

For all j, the sequence $(F_{j,n})$ is computed until the difference between two successive iterates has no exact significant digit. This is not achieved at the same iteration of all values of j. Let n_j be the iteration at which $F_{j,n_j-1} - F_{j,n_j} = @.0$. According to section 2.3, for all j, the exact significant bits of F_{j,n_j} are in common with f_j, up to one.

Let $G_m = \sum_{j=0}^{m} F_{j,n_j}$. According to theorem 1, the exact significant bits of G_m are in common with g_m, up to one.

Figure 2.2 presents the significant bits of G_m and G_{m+1} if the difference $G_m - G_{m+1}$ has no exact significant digit. In this case, the exact significant bits of G_{m+1} are common to G_m and are also common to g_m and g_{m+1}, up to one.

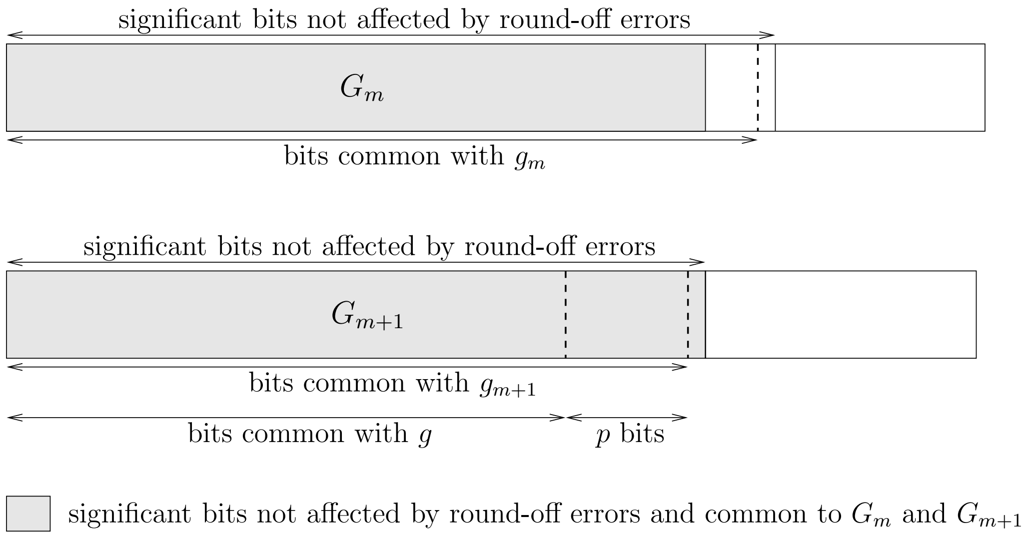

Figure 2.2: Significant bits of G_m and G_{m+1}

We assume that the sequence (g_m) converges at least linearly to g. According to theorem 2, if the convergence zone is reached, $C_{g_m,g_{m+1}} = C_{g_m,g} + \delta$ where δ represents p bits. Therefore the bits common to g_m and g_{m+1} are common with g, up to p.

Consequently if the difference $G_m - G_{m+1}$ has no significant digit, the exact significant bits of G_{m+1} are common with g, up to $p + 1$.

In sections 5.3 and 5.4, we describe the dynamical control of the computation of one-dimensional integrals on an infinite domain.

2.5 Conclusion

In this chapter, we have shown how to dynamically control the computation of sequences of scalar values. An important perspective consists in extending this work to sequences of vectors. Although several studies were dedicated to the dynamical control of iterative methods for solving linear systems, there is still much to do in this direction.

M. Montagnac, for his PhD thesis [67], worked on the dynamical control of several iterative methods: BICGSTAB, CGS and GMRES [74]. The strategies he proposed all use DSA and are based on dynamical restarts of these methods.
I currently supervise with J.-M. Chesneaux the PhD work of R. Adout. We study the acceleration of the convergence, with extrapolation algorithms of the restarted GMRES method. A hybrid method, based on VEA (Vector Epsilon Algorithm) [12] has been proposed [3]. DSA is used to control the numerical quality of this novel method.
Several parameters are used in iterative methods for solving linear systems. In the case of the restarted GMRES method, these parameters are the dimension of the Krylov subspace on which projections are performed and the number of restarts. An important work must be done to determine how to perform a dynamical control of the parameters inherent to each method.

Chapter 3

Dynamical control of multiple integrals computation

3.1 Several approaches for the computation of multiple integrals

Multiple integrals arise in many application fields, such as mechanics or finance. They can be computed using several approaches we present below.

Let us consider the N-dimensional integral

$$I[f] = \int_{\Omega} w(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \tag{3.1}$$

with $\Omega \subset I\!R^N$ and $\forall \mathbf{x} \in \Omega$, $w(\mathbf{x}) \geq 0$.

This integral can be approximated by a weighted sum of integrand evaluations:

$$Q[f] = \sum_{j=1}^{\nu} a_j f(\mathbf{x_j}), \tag{3.2}$$

with $a_j \in I\!R$ and $\mathbf{x_j} \in \Omega$.

The approximation Q is called *quadrature formula* if $N = 1$ and *cubature formula* if $N \geq 2$.

As the approximation (3.2) is assumed to be exact if f is a constant function,

$$I[1] = \int_{\Omega} w(\mathbf{x}) d\mathbf{x} = \sum_{j=1}^{\nu} a_j = Q[1]. \tag{3.3}$$

One can distinguish two classes of cubature formulas: polynomial-based methods and Monte Carlo methods. The choice of these methods depends on the dimension of the integral. In 1984, P.J. Davis and P. Rabinowitz proposed a classification of integrals with respect to their dimension [38]. In 2002, R. Cools considered that this classification could apply almost unchanged [31]. For integrals of dimension less than 8, one can use methods based on formulas which are exact for some polynomials. Monte Carlo or quasi-Monte Carlo methods [15, 68, 69] become competitive for integrals of dimension 8 to about 15 and are particularly indicated for integrals of higher dimension. As this chapter does not deal with the computation of integrals of very high dimension, we describe below polynomial-based methods.

A criterion to classify polynomial-based cubature methods is their degree. A cubature formula is of (polynomial algebraic) degree d if it is exact for polynomials of degree at most d. A multiple integral can be computed with the principle of "iterated integrals" [76]. The integral is expressed as several one-dimensional integrals, which can be computed using a quadrature rule. The principle of "iterated integrals" is equivalent to a cartesian product of several one-dimensional regions [40]. Indeed let us consider the following quadrature rules, respectively of degree d_1 and d_2, for the approximation of integrals on one-dimensional regions R_1 and R_2.

$$\int_{R_1} w_1(x) f_1(x) dx \approx \sum_{i=1}^{\nu_1} a_i f_1(x_i), \tag{3.4}$$

with $\forall i = 1, ..., \nu_1$, $a_i \in I\!R$ and $x_i \in R_1$.

$$\int_{R_2} w_2(y) f_2(y) dy \approx \sum_{j=1}^{\nu_2} a_j f_2(y_j), \tag{3.5}$$

with $\forall j = 1, ..., \nu_2$, $a_j \in I\!R$ and $y_j \in R_2$.

From these two quadrature rules, one can obtain a cubature rule for the approximation of

$$I = \int_{R_1 \times R_2} w_1(x) w_2(y) f(x, y) dx dy. \tag{3.6}$$

The integral I can be expressed as

$$I = \int_{R_1} w_1(x) f_1(x) dx, \tag{3.7}$$

with $\forall x \in R_1$, $f_1(x) = \int_{R_2} w_2(y) f(x, y) dy$.

$$I \approx \sum_{i=1}^{\nu_1} a_i \int_{R_2} w_2(y) f(x_i, y) dy. \tag{3.8}$$

Therefore

$$I \approx \sum_{i=1}^{\nu_1} \sum_{j=1}^{\nu_2} a_i a_j f(x_i, y_j). \tag{3.9}$$

Then the cubature rule (3.9) is of degree $d = \min(d_1, d_2)$ [61].

Using directly a cubature rule such as (3.9) or computing several one-dimensional integrals with the principle of "iterated integrals" can be mathematically equivalent, but leads to different implementations.

Let $I\!P_d^N$ be the vector subspace of all polynomials in N variables of degree at most d. By definition, for a cubature rule of degree d, if the polynomials f_i form a basis for $I\!P_d^N$,

$$Q[f_i] = I[f_i] \ (i = 1, 2, ..., \dim I\!P_d^N). \tag{3.10}$$

$$\sum_{j=1}^{\nu} a_j f_i(\mathbf{x_j}) = I[f_i] \ (i = 1, 2, ..., \dim I\!P_d^N). \tag{3.11}$$

Therefore the weights a_j and the points $\mathbf{x_j}$ in a cubature formula can be determined from the system of non linear equations (3.11).

We know nowadays that, for a given degree, the minimal number of points depends on the region of integration and on the weight function w [31]. However we still do not know the minimal number of points for simple regions such as a square or a triangle with a constant weight function.

According to [35], a significant collection of cubature formulas have been established for only four two-dimensional regions: the square, the circle, the unitary triangle and the plane. However, thanks to affine transformations which do not affect their degree, these cubature formulas can be adapted to other two-dimensional regions such as a parallelogram or an ellipse.

Cubpack is a library for the computation of multiple integrals. A first version has been written in Fortran 77 in 1992 [32]. Then a C++ version, named Cubpack++, has been developed [34]. The Cubpack++ library enables the computation of integrals over two-dimensional regions which can be defined from the four basic regions previously described. Cubature formulas of degree 13 with 36 or 37 points are used with an adaptive strategy. A new version of the Cubpack library has been written in Fortran 95 [33]. It enables the computation of integrals of higher dimension. It also proposes an extrapolation procedure using the ε-algorithm for non triangular regions. The Cubpack library always provides an estimation of the truncation error.

The algorithms developed in interval arithmetic for the computation of multiple integrals take into account both the truncation error and the round-off error. The cubature methods used are usually based on the product of quadrature rules. Adaptive strategies have been established and use, for instance, Romberg's method [78], Clenshaw-Curtis method [17], or Gaussian quadrature methods [17, 43, 52, 53, 89]. Two libraries have been developed in interval arithmetic for the computation of integrals of dimension 1 or 2. VANI (Validated Adaptive Numerical Integration) [17] has been written in Pascal-XSC. Then CLAVIS (CLAsses for Verified Integration over Singularities) [89] has been developed in C-XSC and can be accessed on the web site dedicated to XSC languages[1].

The strategies for the computation of multiple integrals, which are described in the following sections, are based on partitions of the integration domain. With a dynamical control of the number of partitions, one can compute an approximation which minimizes the global error, consisting of both the truncation error and the round-off error. We show how to determine

[1] `http://www.xsc.de`

the significant digits of the result obtained which are in common with the exact value of the integral. These strategies have been implemented in the ACIM (Atelier de Calcul d'Intégrales Multiples) library, which enables the computation of integrals of dimension 1 to 3 using DSA (Discrete Stochastic Arithmetic).

3.2 Approximation using the principle of "iterated integrals"

The principle of "iterated integrals" [76] consists in expressing a multiple integral as several one-dimensional integrals, which can be computed using a quadrature method. The strategies described below are based on the theoretical results presented in 2.3. They enable one to compute the different one-dimensional integrals with an optimal integration step. Furthermore one can determine which digits in the approximation obtained are in common with the exact value of the integral. In the strategies described in this section, small letters denote exact values and capital letters the corresponding approximations computed using DSA.

3.2.1 Computation of two-dimensional integrals

Let f be a continuous function on a two-dimensional finite domain D. Let us consider the following integral.

$$s = \int_D f(x,y)dxdy = \int_a^b \int_{y_1(x)}^{y_2(x)} f(x,y)dxdy. \tag{3.12}$$

The integral s can be expressed as one-dimensional integrals:
$s = \int_a^b g(x)dx$, with $\forall x \in [a,b],\ \ g(x) = \int_{y_1(x)}^{y_2(x)} f(x,y)dy$.

Therefore the integral s can be evaluated using a quadrature method, such as the trapezoidal rule, Simpson's rule, Romberg's method or the Gauss-Legendre method.

The approximation of s requires the evaluation of the integral $g(x_i)$ for different points x_i in $[a,b]$. Each integral $g(x_i)$ can be evaluated using one of the strategies described in 2.3. Let $G_m(x_i)$ be the approximation of $g(x_i)$ computed:

- using the trapezoidal rule or Simpson's rule with the integration step $\frac{y_2(x_i)-y_1(x_i)}{2^m}$,
- by partitioning the integration interval $[y_1(x_i), y_2(x_i)]$ in 2^m subintervals on which the Gauss-Legendre method is used
- or by performing m iterations of Romberg's method.

The sequence $(G_m(x_i))$ is computed until the difference between two successive iterates has no exact significant digit. This condition is not necessarily satisfied at the same iteration for each point x_i. Let m_i be the iteration at which the difference $G_{m_i-1}(x_i) - G_{m_i}(x_i)$ has no exact significant digit. From the theoretical results described in 2.3, the exact significant bits of $G_{m_i}(x_i)$ are in common with $g(x_i)$, up to δ with

$$\delta = \log_2\left(\frac{2^p}{2^p-1}\right), \tag{3.13}$$

p being the order of the quadrature method ($p = 2$ for the trapezoidal rule, $p = 4$ for Simpson's rule, $p = 2\nu$ for the Gauss-Legendre method with ν points) and $\delta = 0$ for Romberg's method. δ represents at most one bit.

Let S_n be the approximation of s computed with these values $G_{m_i}(x_i)$ using one of the strategies described in 2.3. Therefore the approximation S_n can be obtained:

- using the trapezoidal rule or Simpson's rule with the integration step $\frac{b-a}{2^n}$,
- by partitioning the integration interval in 2^n subintervals on which the Gauss-Legendre method is used
- or by performing n iterations of Romberg's method.

S_n is a linear combination of approximations $G_{m_i}(x_i)$: $S_n = \phi(\{G_{m_i}(x_i)\})$, the function ϕ depending on the integration method which is chosen. Let us denote by s_n the value obtained by using the same quadrature method with the exact values $g(x_i)$: $s_n = \phi(\{g(x_i)\})$.
Since S_n is a linear combination of the approximations $G_{m_i}(x_i)$ the exact significant bits of which being in common with the exact values $g(x_i)$ up to δ, from theorem 1, the exact significant bits of S_n are in common with s_n up to δ.
From the theoretical results described in 2.3, if the convergence zone of the sequence (s_n) is reached, the significant bits common to s_{n-1} and s_n are also common to s, up to δ.

If the difference between two successive approximations S_{n-1} and S_n has no exact significant digit, all the exact significant bits of S_n are in common with S_{n-1}. Since they are also in common with s_n and s_{n-1} up to δ, they are in common with s, up to 2δ.
Figure 3.1 presents the significant bits of S_{n-1} and S_n if their difference has no exact significant digit.

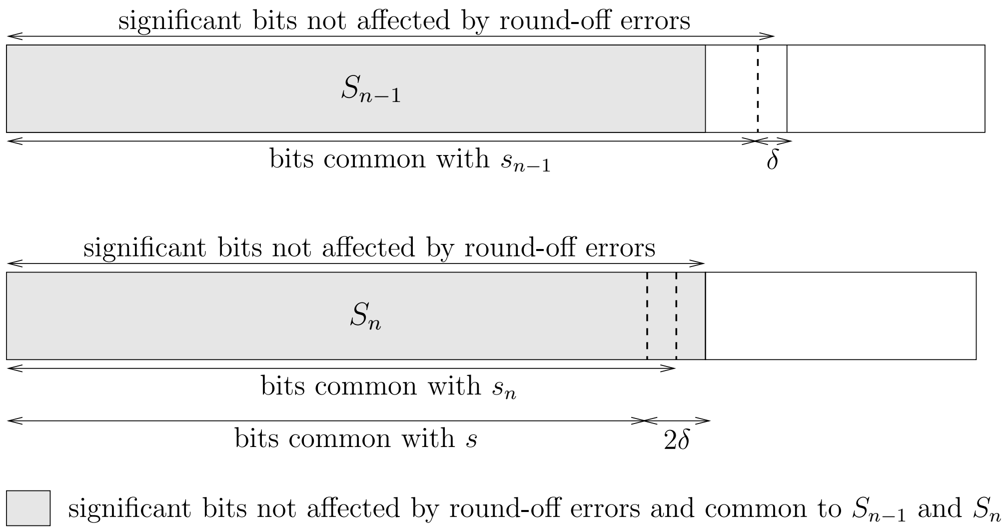

Figure 3.1: Significant bits of S_{n-1} and S_n if $S_{n-1} - S_n = @.0$

Whatever the quadrature method chosen, 2δ represents at most one bit. Therefore the exact significant bits of the approximation S_n are in common with s, the exact value of the integral, up to one.

3.2.2 Computation of three-dimensional integrals

Let f be a continuous function on a three-dimensional finite domain D. Let us consider the following integral.

$$s = \int_D f(x,y,z)\,dx\,dy\,dz = \int_a^b \int_{y_1(x)}^{y_2(x)} \int_{z_1(x,y)}^{z_2(x,y)} f(x,y,z)\,dx\,dy\,dz. \tag{3.14}$$

The integral s can be expressed as:
$s = \int_a^b h(x)dx$, where $\forall x \in [a,b],\ h(x) = \int_{y_1(x)}^{y_2(x)} \int_{z_1(x,y)}^{z_2(x,y)} f(x,y,z)\,dy\,dz$.

The approximation of s requires the evaluation of two-dimensional integrals $h(x_i)$ with $x_i \in [a,b]$.
Let $H_{m_i}(x_i)$ be the approximation of $h(x_i)$ computed using the strategy previously described. Therefore the exact significant bits of $H_{m_i}(x_i)$ are in common with $h(x_i)$, up to 2δ.
Let S_n be the approximation of s computed with these values $H_{m_i}(x_i)$ using one of the strategies described in 2.3. Therefore the approximation S_n can be obtained:

- using the trapezoidal rule or Simpson's rule with the integration step $\frac{b-a}{2^n}$,
- by partitioning the integration interval in 2^n subintervals on which the Gauss-Legendre method is used
- or by computing n iterations of Romberg's method.

S_n is a linear combination of approximations $H_{m_i}(x_i)$: $S_n = \phi(\{H_{m_i}(x_i)\})$, the function ϕ depending on the integration method which is chosen.
Let us denote by s_n the value obtained by using the same quadrature method with the exact values $h(x_i)$: $s_n = \phi(\{h(x_i)\})$.
Since S_n is a linear combination of the approximations $H_{m_i}(x_i)$ the exact significant bits of which being in common with the exact values $h(x_i)$ up to 2δ, from theorem 1, the exact significant bits of S_n are in common with s_n up to 2δ.
From the theoretical results described in 2.3, if the convergence zone of the sequence (s_n) is reached, the significant bits common to s_{n-1} and s_n are also common to s, up to δ.

If the difference between two successive approximations S_{n-1} and S_n has no exact significant digit, all the exact significant bits of S_n are in common with S_{n-1}. Since they are also in common with s_n and s_{n-1} up to 2δ, they are in common with s, up to 3δ.
Figure 3.2 presents the significant bits of S_{n-1} and S_n if their difference has no exact significant digit.

If the trapezoidal rule is used, $3\delta = 3\log_2\left(\frac{4}{3}\right) \approx 1.2$. In this case, the exact significant bits of the approximation S_n are in common with s, the exact value of the integral, up to two.
With the other quadrature methods, 3δ represents at most one bit: the exact significant bits of S_n are in common with s, up to one.

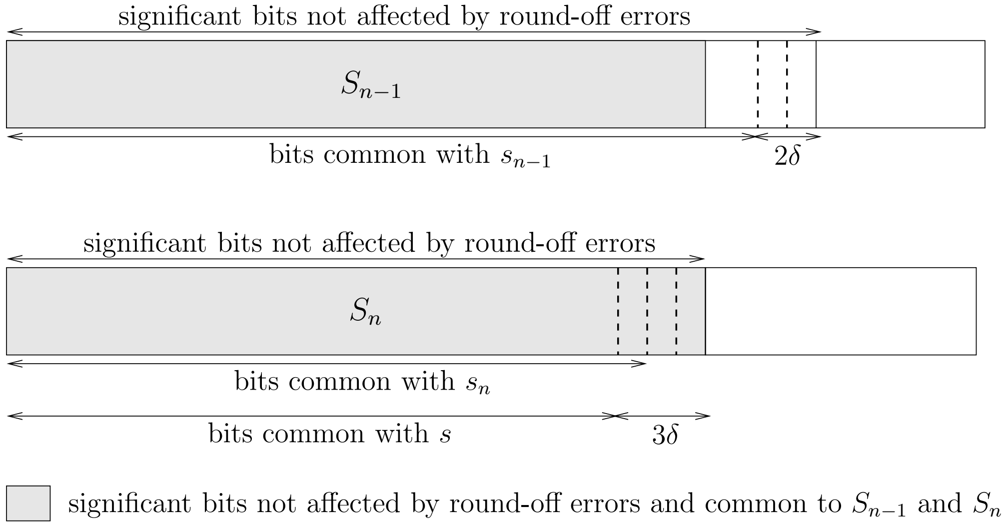

Figure 3.2: Significant bits of S_{n-1} and S_n if $S_{n-1} - S_n = @.0$

3.2.3 Computation of N-dimensional integrals

The strategies described for two-dimensional and three-dimensional integrals can be generalized to integrals on an N-dimensional finite domain. The exact significant bits of the approximation obtained are in common with the exact value of the integral, up to $N\delta$.

With Romberg's method, $\delta = 0$.

With the other quadrature methods which have been studied, $N\delta = \log_2\left(\left(\frac{2^p}{2^p-1}\right)^N\right)$, p being the order of the method. Let us evaluate M_p, the maximal value of N such that $\log_2\left(\left(\frac{2^p}{2^p-1}\right)^N\right) < 1$.

With the trapezoidal rule, $p = 2$ and $M_p = 2$. However if $N \leq 8$, $\log_{10}\left(\left(\frac{2^p}{2^p-1}\right)^N\right) < 1$. Therefore, if the trapezoidal rule is used, one can obtain an approximation of an integral of dimension 1 to 8, the exact significant decimal digits of which being common with the exact value, up to one.

With Simpson's rule, $p = 4$ and $M_p = 35$. Therefore, if Simpson's rule is used, one can obtain an approximation of an integral of dimension 1 to 35, the exact significant bits of which being common with the exact value, up to one.

The value M_p increases strongly as the order p increases. With the Gauss-Legendre method with 6 points, $p = 12$ and $M_p = 2838$.

3.3 Approximation using cubature methods

Cubature methods enable the computation of a multiple integral using a formula which is adapted to the integration domain. In this section, we show how to perform a dynamical control of cubature methods on the hypercube $[0,1]^N$. Then the strategy described below will be generalized to the computation of integrals on an N-dimensional hyperrectangle.

First let us briefly review some properties of cubature formulas for the approximation of integrals on the hypercube $[0,1]^N$. These properties are given in several publications, *i.e.* [60, 82] and [61] for the two-dimensional case.

Let

$$I = \int_{[0,1]^N} f(\mathbf{x}) d\mathbf{x}. \tag{3.15}$$

This integral can be approximated using a cubature formula Q with ν points in accordance with equation (3.2). As the cubature formula Q is assumed to be exact if the function f is constant, the weights a_j satisfy:

$$\sum_{j=1}^{\nu} a_j = 1. \tag{3.16}$$

By partitioning the hypercube $[0,1]^N$ in m^N hypercubes of side length $\frac{1}{m}$ and by applying on each one a properly scaled version of Q, one obtains the formula $Q^{(m)}$, named "m^N-copy rule" in for example [60, 61]:

$$Q^{(m)}[f] = \frac{1}{m^N} \sum_{\mathbf{k} \in \{0,1,\ldots,m-1\}^N} \sum_{j=1}^{\nu} a_j f\left(\frac{\mathbf{x_j} + \mathbf{k}}{m}\right). \tag{3.17}$$

Theorem 10 presents the Euler-Maclaurin expansion of $Q^{(m)} f - I$. It has been established in [60], from several results described in [9, 59, 63].

Theorem 10 *Let I and $Q^{(m)}[f]$ be defined by equations (3.15) and (3.17).*
Let us assume that all partial derivatives of f of total order at most p are integrable over $[0,1]^N$, then

$$Q^{(m)}[f] - I = \sum_{s=1}^{l-1} \frac{B_s}{m^s} + \mathcal{O}\left(\frac{1}{m^l}\right), \quad N \le l \le p$$

where the coefficients B_s depends on Q and f, but not on m.

The coefficients B_s satisfy the following equation

$$B_s = \sum_{|\mathbf{s}|=s} c_{\mathbf{s}}(Q) \int_{[0,1]^N} f^{(\mathbf{s})}(\mathbf{x}) d\mathbf{x} \tag{3.18}$$

with

$$|\mathbf{s}| = |(s_1, s_2, \ldots, s_N)| = \sum_{i=1}^{N} s_i. \tag{3.19}$$

The coefficients $c_{\mathbf{s}}(Q)$ may be obtained by applying the rule Q to a product of Bernoulli polynomials $B_q(x)$. Thus,

$$c_{\mathbf{s}}(Q) = Q[\phi], \quad \text{with} \quad \phi(\mathbf{x}) = \Pi_{i=1}^{N} \frac{B_{s_i}(x_i)}{s_i!}. \tag{3.20}$$

Properties :

1. If the cubature formula Q is of (polynomial algebraic) degree d, *i.e.* exact for any polynomial of degree at most d, then $B_s = 0$, $s = 1, 2, ..., d$.

2. If the cubature formula Q is symmetric, *i.e.* invariant by symmetry with respect to $x_i = \frac{1}{2}$ $(i = 1, 2, ..., N)$, then $B_s = 0$, $\forall s$ odd. The Euler-Maclaurin expansion of $Q^{(m)}[f] - I$ contains only odd powers of $\frac{1}{m}$.

3. If f is a polynomial of degree $d(f)$, then $B_s = 0$, $\forall s > d(f)$.

Corollary 1 can be deduced from theorem 10. Let us consider a sequence of approximations I_n obtained by partitioning the integration domain $[0, 1]^N$ in hypercubes of side length $\frac{1}{2^n}$ and using the same cubature method on each subdomain.

Corollary 1 *Let the integral I and the cubature formula Q be defined by equations (3.15) and (3.2).*
Let I_n be the approximation of I obtained by partitioning the hypercube $[0, 1]^N$ in hypercubes of side length $\frac{1}{2^n}$ and applying on each one the cubature formula Q.
If Q is of degree d, then

$$I_n - I = \frac{B_{d+1}}{2^{n(d+1)}} + \mathcal{O}\left(\frac{1}{2^{n(d+2)}}\right),$$

B_{d+1} being defined by equation (3.18).

Proof Since $I_n = Q^{(2n)}[f]$, the Euler-Maclaurin expansion of $I_n - I$ can be deduced from theorem 10 with $m = 2^n$.
Furthermore, since Q is of degree d, from the first property previously given, the first d terms in this expansion vanish. □

Theorem 11 enables one, from two successive approximations I_n and I_{n+1}, to determine the first digits of the exact value of the integral. The same notations and assumptions as in corollary 1 are used.

Theorem 11

$$C_{I_n, I_{n+1}} = C_{I_n, I} + \log_{10}\left(\frac{2^{d+1}}{2^{d+1} - 1}\right) + \mathcal{O}\left(\frac{1}{2^n}\right).$$

The proof of theorem 11 is similar to the one of theorem 4 concerning approximation methods. If $K = B_{d+1}$, $p = d + 1$ and $q = d + 2$ in equation (2.1), from which theorem 4 has been established, one obtains corollary 1.

If the convergence zone is reached, *i.e.* if $\mathcal{O}\left(\frac{1}{2^n}\right)$ becomes negligible, the significant bits common to two successive approximations are also in common with the exact value of the integral, up to one. Indeed, whatever the degree d of the cubature method, $\log_{10}\left(\frac{2^{d+1}}{2^{d+1}-1}\right)$ corresponds to one bit.
If approximations I_n are computed until, in the convergence zone, the difference $I_n - I_{n+1}$ has no exact significant digit, we can guarantee that the exact significant bits of the last iterate are in common with I, up to one.

The strategy described for the computation of integrals on the hypercube $[0,1]^N$ can be generalized to the computation of integrals of the form

$$J = \int_D g(\mathbf{y})d\mathbf{y}, \tag{3.21}$$

where $D = [a_1, b_1] \times [a_2, b_2] \times ... \times [a_N, b_N]$.
Indeed, by a change of variable, J can be expressed as an integral on the hypercube $[0,1]^N$.

$$J = \Pi_{i=1}^N (b_i - a_i) I = \Pi_{i=1}^N (b_i - a_i) \int_{[0,1]^N} f(\mathbf{x})d\mathbf{x}, \tag{3.22}$$

where $f(\mathbf{x}) = g\left((b_1 - a_1)x_1 + a_1,\ (b_2 - a_2)x_2 + a_2, ...,\ (b_N - a_N)x_N + a_N\right)$.
Adopting the same notations as in corollary 1, one can approximate the integral J by

$$J_n = \Pi_{i=1}^N (b_i - a_i) I_n. \tag{3.23}$$

Then the truncation error on J_n is

$$J_n - J = \Pi_{i=1}^N (b_i - a_i)(I_n - I) \tag{3.24}$$

where $I_n - I$ is given in corollary 1. As the term $\Pi_{i=1}^N (b_i - a_i)$ depends neither on the cubature formula used, nor on the number of partitions of the integration domain, the equation established in theorem 11 is also satisfied by the sequence (J_n).

3.4 The ACIM library

The ACIM (Atelier de Calcul d'Intégrales Multiples) library enables the computation using DSA in double precision of integrals of dimension 1 to 3 with the strategies described in chapters 2 and 3. The ACIM library is a collection of functions written in Fortran 90 which can be used to compute integrals with fixed or variable, finite or infinite bounds.

Three quadrature methods have been implemented: Simpson's rule, Romberg's method and the Gauss-Legendre method with 12 points. They can be used to compute multiple integrals with the principle of "iterated integrals".

Two-dimensional integrals with fixed and finite bounds can also be computed using six cubature methods, denoted by C^1_Ω to C^6_Ω. These methods consist in evaluating an integral on the square $\Omega = [-1,1]^2$ by using ν points in this square. Thus an integral

$$I_\Omega[f] = \int_\Omega f(x,y)\,dx\,dy \tag{3.25}$$

is approximated by

$$C_\Omega[f] = \sum_{i=1}^{\nu} a_i f(x_i, y_i) \tag{3.26}$$

with for $i = 1, ..., \nu$, $a_i \in I\!R$ and $(x_i, y_i) \in \Omega$. A change of variable can be performed for the approximation of integrals on other two-dimensional domains. Features of the methods C^1_Ω to C^3_Ω, which can be consulted in [1], are reported in table 3.1. Figure 3.3 presents, for the methods C^1_Ω to C^3_Ω, the position of the ν points (x_i, y_i) in the square Ω. The methods C^4_Ω, C^5_Ω and C^6_Ω, respectively of degree 13, 15 or 19, correspond to the Gauss-Legendre method, with respectively 7, 8 or 10 points, applied to each side of the integration domain.

method	degree	ν	(x_i, y_i)	a_i
C^1_Ω	3	9	$(0,0)$	16/9
			$(0,\pm 1)$	4/9
			$(\pm 1, 0)$	4/9
			$(\pm 1, \pm 1)$	1/9
C^2_Ω	3	4	$(\pm\sqrt{\frac{1}{3}}, \pm\sqrt{\frac{1}{3}})$	1
C^3_Ω	5	9	$(0,0)$	64/81
			$(\pm\sqrt{\frac{3}{5}}, \pm\sqrt{\frac{3}{5}})$	25/81
			$(\pm\sqrt{\frac{3}{5}}, 0)$	40/81
			$(0, \pm\sqrt{\frac{3}{5}})$	40/81

Table 3.1: Features of the methods C^1_Ω to C^3_Ω

Three cubature methods have been implemented in ACIM for the computation of three-dimensional integrals with fixed and finite bounds. These methods, denoted by C^1_Γ to C^5_Γ, consist in evaluating an integral on the cube $\Gamma = [-1,1]^3$ by using ν points in this cube. Thus an integral

$$I_\Gamma[f] = \int_\Gamma f(x,y,z)\,dx\,dy\,dz \tag{3.27}$$

is approximated by

$$C_\Gamma[f] = \sum_{i=1}^{\nu} a_i f(x_i, y_i, z_i) \tag{3.28}$$

with for $i = 1, ..., \nu$, $a_i \in I\!R$ and $(x_i, y_i, z_i) \in \Gamma$. A change of variable can be performed for the approximation of integrals on other three-dimensional domains. Features of the methods

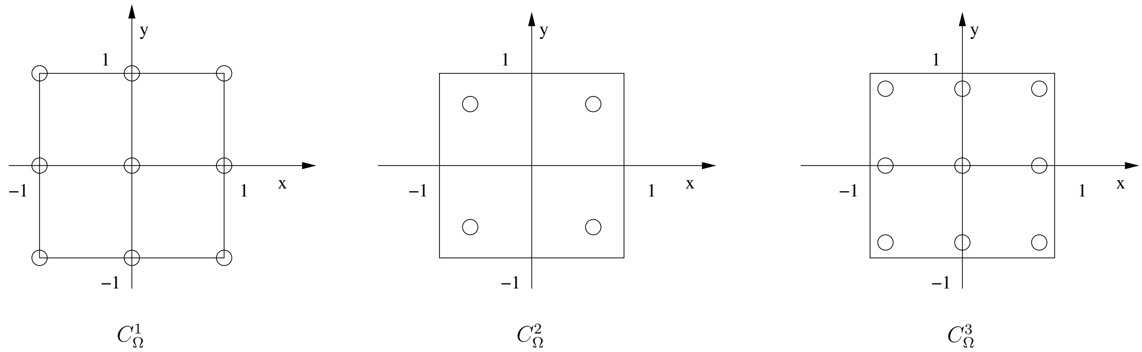

Figure 3.3: Position of the ν points (x_i, y_i) in the square Ω

C^1_Γ to C^3_Γ, which can be consulted in [1], are reported in table 3.2. Figure 3.4 presents, for the methods C^1_Γ and C^2_Γ, the position of the ν points (x_i, y_i, z_i) in the cube Γ. Methods C^4_Γ and C^5_Γ, respectively of degree 9 or 19, correspond to the Gauss-Legendre, with respectively 5 or 10 points, to each side of the integration domain.

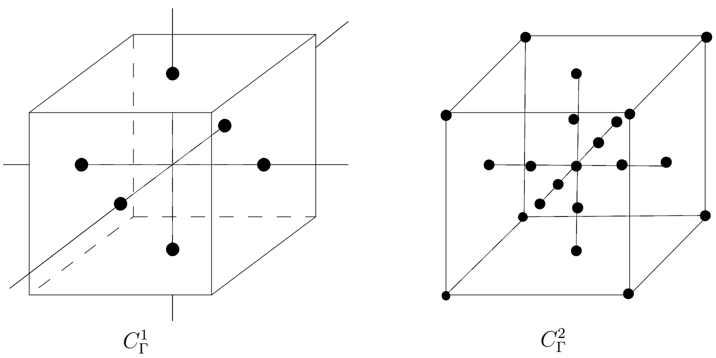

Figure 3.4: Position of the ν points (x_i, y_i, z_i) in the cube Γ

Whatever the integration method which is chosen, a sequence of approximations is computed in DSA until the difference between two successive iterates has no exact significant digit. From the theoretical results previously described, one can determine the significant digits of the approximation obtained which are in common with the exact value of the integral.

The numerical experiment described below has been carried out using the ACIM library. Let us consider the following integrals:

1. $\int_0^1 \int_0^1 y \frac{arctan(\sqrt{2+x^2})}{(1+x^2)\sqrt{2+x^2}} dxdy = \frac{5\pi^2}{192}$,

2. $\int_0^1 \int_0^1 xy \log(1+y) dxdy = \frac{1}{8}$,

3. $\int_0^1 \int_0^1 \int_0^1 ye^{-\frac{x}{2}} e^{-z} dxdydz = (1 - \frac{1}{e})(1 - \frac{1}{\sqrt{e}})$,

4. $\int_0^1 \int_0^1 \int_0^1 z \frac{e^{-y}}{1+x} dxdydz = (1 - \frac{1}{e}) \frac{log(2)}{2}$.

method	degree	ν	(x_i, y_i, z_i)	a_i
C_Γ^1	3	6	$(\pm 1, 0, 0)$	1/6
			$(0, \pm 1, 0)$	1/6
			$(0, 0, \pm 1)$	1/6
C_Γ^2	5	21	$(0, 0, 0)$	-496/45
			$(\pm\frac{1}{2}, 0, 0)$	128/45
			$(0, \pm\frac{1}{2}, 0)$	128/45
			$(0, 0, \pm\frac{1}{2})$	128/45
			$(\pm 1, 0, 0)$	8/45
			$(0, \pm 1, 0)$	8/45
			$(0, 0, \pm 1)$	8/45
			$(\pm 1, \pm 1, \pm 1)$	5/45
C_Γ^3	7	42	$(\pm 1, 0, 0)$	728/450
			$(0, \pm 1, 0)$	728/450
			$(0, 0, \pm 1)$	728/450
			$(0, \pm 1, \pm 1)$	320/450
			$(\pm 1, \pm 1, 0)$	320/450
			$(\pm 1, 0, \pm 1)$	320/450
			$(\pm 1, \pm\frac{\sqrt{10}}{4}, \pm\frac{\sqrt{10}}{4})$	128/450
			$(\pm\frac{\sqrt{10}}{4}, \pm 1, \pm\frac{\sqrt{10}}{4})$	128/450
			$(\pm\frac{\sqrt{10}}{4}, \pm\frac{\sqrt{10}}{4}, \pm 1)$	128/450

Table 3.2: Features of the methods C_Γ^1 to C_Γ^3

Table 3.3 presents the approximations obtained with ACIM and the 16 first digits of the exact value of each integral. Run times have been measured on a 500 Mhz Pentium III processor and are expressed in seconds. The integrals have been computed, on the one hand using the principle of "iterated integrals" with Simpson's rule, Romberg's method or the Gauss-Legendre method with 12 points, on the other hand using the cubature methods previously described. For each result obtained, only the exact significant digits (*i.e.* not affected by round-off errors) are reported in table 3.3.

One can notice that according to the theoretical results previously described, the exact significant decimal digits of the approximations obtained are in common with the mathematical value of the integral, up to one.

The computation with Romberg's method or the Gauss-Legendre method with 12 points applied to each side of the integration domain, with the principle of "iterated integrals", is both reliable and powerful.

Concerning the two-dimensional integrals, the cubature method C_Ω^6, of degree 19, provides comparable results in terms of run time and numerical quality. For the integral 1, one can notice that the methods C_Ω^5 and C_Ω^6 both provide a result with the maximal accuracy in double precision (15 exact significant digits). Run time is a little longer with the method C_Ω^6, which requires more evaluations of the integrand.

Concerning the three-dimensional integrals, besides Romberg's method and the Gauss-Legendre method applied three times, the cubature methods C_Γ^4 and C_Γ^5, respectively of degree 9 or 19, enable reliable and powerful computation of the integrals which have been studied. For the integral 3, the approximation obtained with the method C_Γ^4 (with 125 points) has the maximal accuracy (15 exact significant digits); the approximation obtained with the method C_Γ^5 (with 1000 points) has one less exact significant digit. On the contrary, for integral 4, the method C_Γ^5 is more indicated than the method C_Γ^4: it provides an approximation with the maximal accuracy in a relatively low run time.
The principle of "iterated integrals" can be worthwhile in DSA, because it enables one to determine the optimal integration step for each dimension of the domain.

In section 5.5, we describe the dynamical control of the computation of a three-dimensional improper integral involved in the neutron star theory.

3.5 Conclusion

By performing a dynamical control of the computation of a multiple integral, like with the numerical methods described in the previous chapter, one can obtain an approximation which minimizes the global error. In the case of a non singular integral on a finite region, the exact significant digits (*i.e* not affected by round-off errors) in the approximation obtained are usually in common with the mathematical value, up to one.

The strategies described in section 3.2 enable one, with the principle of "iterated integrals" to dynamically control the step value for each dimension of the integration domain. Then one can determine the number of exact significant bits of the result which are in common with mathematical value of the integral. This number depends on the dimension of the integration domain, on the number of improper bounds, on the possible approximation of the integrand by its series expansion, on the convergence speed of the required sequences. Therefore a specific study of the integral, such as the one described in section 5.5, is sometimes necessary.

For the computation of a non singular integral on a hyperrectangle, the strategy described in section 3.3 is based on the use of a cubature method on subregions of the same size. The optimal number of partitions can be dynamically controlled. From the Euler-Maclaurin expansion of the truncation error, we have shown that the exact significant bits of the approximation obtained are in common with the mathematical value of the integral, up to one. This result could be extended to other types of integrals using the expansion of the truncation error generated. For instance, the Euler-Maclaurin expansion has been established for N-dimensional simplexes [64]. Furthermore several studies were dedicated to the truncation error expansion generated by the computation of singular integrals [60, 62, 75, 81].

integral	method	approximation	run time (s)
	exact value	0.257020947945035E +000	-
1	Simpson	0.25702094794503E +000	8.0E+00
	Romberg	0.257020947945035E +000	2.1E-02
	Gauss-Legendre	0.257020947945035E +000	1.9E-02
	C^1_Ω	0.2570209479450E +000	1.0E+01
	C^2_Ω	0.25702094794503E +000	1.7E+01
	C^3_Ω	0.25702094794503E +000	2.3E+00
	C^4_Ω	0.257020947945035E +000	5.E-02
	C^5_Ω	0.257020947945035E +000	2.0E-02
	C^6_Ω	0.257020947945035E +000	3.0E-02
	exact value	0.125000000000000E +000	-
2	Simpson	0.124999999999999E+000	8.9E-02
	Romberg	0.124999999999999E+000	1.0E-03
	Gauss-Legendre	0.124999999999999E+000	1.9E-03
	C^1_Ω	0.1250000000000E+000	1.6E+01
	C^2_Ω	0.1249999999999E+000	3.4E+01
	C^3_Ω	0.1250000000000E+000	1.0E+00
	C^4_Ω	0.125000000000000E +000	2.5E-02
	C^5_Ω	0.124999999999999E +000	8.2E-03
	C^6_Ω	0.125000000000000E +000	3.0E-03
	exact value	0.248720059264354E+000	-
3	Simpson	0.24872005926435E+000	6.4E+01
	Romberg	0.248720059264354E+000	6.9E-02
	Gauss-Legendre	0.248720059264354E+000	3.9E-01
	C^1_Γ	0.2487200592642E+000	1.1E+03
	C^2_Γ	0.2487200592643E+000	4.1E+01
	C^3_Γ	0.2487200592643E+000	8.2E+01
	C^4_Γ	0.248720059264353E+000	7.0E-02
	C^5_Γ	0.24872005926435E+000	7.0E-02
	exact value	0.219076291562995E+000	-
4	Simpson	0.21907629156299E+000	3.2E+02
	Romberg	0.219076291562995E+000	7.0E-01
	Gauss-Legendre	0.219076291562995E+000	3.0E-01
	C^1_Γ	0.21907629156E+000	3.5E+05
	C^2_Γ	0.2190762915629E+000	3.0E+01
	C^3_Γ	0.219076291562E+000	4.7E+02
	C^4_Γ	0.21907629156299E+000	4.0E+00
	C^5_Γ	0.219076291562996E+000	5.0E-02

Table 3.3: Computation of integrals with the ACIM library

Conclusion and perspectives

The strategies described in chapters 2 and 3 are all based on a sequence computation. This sequence can be inherent to the method used. For instance, Romberg's method consists in computing a sequence of approximations of an integral. If a discretization step is used in a numerical method, a sequence of approximations can be generated by decreasing regularly this step. A sequence can also be computed by partitioning successively the domain studied into subdomains on which the same numerical method is used.

The theoretical results established in chapters 2 and 3 enable one, from two successive iterates in the convergence zone, to determine the first digits of the exact limit. These theoretical results become particularly interesting if one can estimate round-off error propagation, for instance thanks to Discrete Stochastic Arithmetic (DSA). If computations are performed until, in the convergence zone, the difference between two successive iterates has no exact significant digit, the global error on the last iterate is minimal. Then one can determine which significant digits in the result obtained are in common with the exact limit.

The different theorems established in this document can combine and thus apply to numerical algorithms requiring the computation of several sequences. Theorem 1, given in chapter 1, is fundamental, because it enables one, if an arithmetical operation is performed between several sequences, to determine the significant digits of the result in common with the corresponding exact value. However these theoretical results do not exempt from a preliminary study of the problem. Indeed information on the convergence of sequences is essential.

The strategies described in chapters 2 and 3 consist in computing each sequence until the difference between two successive iterates has no exact significant digit. The number of iterations required for this stopping criterion to be satisfied and, consequently, the run time depend on the precision chosen (*e.g.* single or double precision in IEEE arithmetic). An integral on an infinite domain can be approximated by a sum of integrals on finite domains. With the strategy described in 2.4.2, the global error on each integral is minimal. However if an integral has an order of magnitude which is very low compared to the total being computed, only its first significant digits are taken into account. Other stopping criteria can be used in order to prevent the computation of some integrals with a needlessly high accuracy. Therefore new theorems related to these stopping criteria should be established in order to determine the significant digits in common between the approximation obtained and the corresponding exact value.

Improvements in automatic methods for round-off error analysis are required because of the performances, always higher, of computers. In order to enable an estimation of round-off error at a moderate computing cost, one has to take into account the evolution of the architectures, the compilers and the processors. Several new implementations of DSA can be considered. A library could be developed for MATLAB codes. A library, which ebables the use of interval arithmetic in MATLAB codes has already been written [41, 73]. A library implementing DSA in Java codes could also be developed. Such an implementation has already been done for interval arithmetic [11]. Another possibility is the development of a compiler which integrates stochastic types and functionalities of DSA. A Fortran 95 compiler already integrates interval arithmetic [80]. In order to allow the use of the random rounding mode, essential for DSA, special routines have been developed for specific processors in assembly language. The implementation of DSA on the Itanium processor is a research project which could benefit from the studies carried out for the implementation of the CENA method on this processor [56].

Linear algebra libraries are widely spread for scientific computing. In the LAPACK library [6], error bounds on the results are computed using inverse analysis. A proposal [36] has been written for the implementation of BLAS (Basic Linear Algebra Subroutines) in interval arithmetic. It would be interesting to develop a linear algebra library that would integrate functionalities of DSA for the estimation of round-off error and the dynamical control of computations.

When grid computing is used for numerical simulation, a code can be run on heterogeneous machines which possibly different arithmetics. The identity of these machines is not always known by the user. It can be difficult to implement an automatic method for round-off error analysis on such a grid. In order to give to the user relevant information about the numerical quality of his code, one should propose new methodologies which could use the possible differences in the result from one execution to another.

Chapter 4

Bibliography

[1] M. Abramowitz and I.A. Stegun. *Handbook of mathematical functions*, volume 55 of *Applied Mathematics Series*. Washington : National Bureau of Standards, 1964. reprinted 1968 by Dover Publications, New York.

[2] E. Adams and U. Kulisch, editors. *Scientific computing with automatic result verification*, volume 189 of *Mathematics in Science and Engineering*. Academic Press, San Diego, 1993.

[3] R. Adout, J.-M. Chesneaux, and F. Jézéquel. A reliable hybrid extrapolation algorithm to accelerate the restarted GMRES method. In *Proc. SCAN2004 conference, Fukuoka, Japan*, October 2004.

[4] G. Alefeld and J. Herzberger. *Introduction to interval analysis*. Academic Press, 1983.

[5] R. Alt. Floating-point error propagation in iterative methods. *Math. Comput. Simulation*, 30(6):505–517, 1988.

[6] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 3rd edition, 1999.

[7] A. Asserrhine. *Implémentation de l'arithmétique aléatoire sur machines distribuées et application*. PhD thesis, Université P. et M. Curie, Paris, France, May 1996.

[8] A. Asserrhine, J.-M. Chesneaux, and J.-L. Lamotte. Estimation of round-off errors on several computers architectures. *J. of Universal Computer Science*, 1(7):454–468, July 1995.

[9] C.T.H. Baker and G.S. Hodgson. Asymptotic expansions for integration formulas in one or more dimensions. *SIAM J. Numer. Anal.*, 8:473–480, 1971.

[10] F.L. Bauer, H. Rutishauser, and E. Stiefel. New aspects in numerical quadrature. In *Proc. Sympos. Appl. Math.*, volume XV, pages 199–218. Amer. Math. Soc., Providence, R.I., 1963.

[11] B.R. Callajas Bedregal and J. Enéas Montenegro Dutra. An intervalar library in Java. In *SCAN2000, 9th International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics, Karlsruhe, Germany*. GAMM-IMACS, September 2000.

[12] C. Brezinski and M. Redivo Zaglia. *Extrapolation Methods, theory and Practice*. North Holland, 1991.

[13] R.L. Burden and J.D. Faires. *Numerical analysis*. Brooks-Cole Publishing, 7th edition, 2001.

[14] CADNA: Control of Accuracy and Debugging for Numerical Applications. Université Pierre et Maris Curie, Paris. `http://www.lip6.fr/cadna`.

[15] R.E. Caflisch. Monte Carlo and Quasi-Monte Carlo Methods. In *Acta Numerica*, volume 7, pages 1–49. Cambridge University Press, Cambridge, 1998.

[16] F. Chaitin-Chatelin and V. Frayssé. *Lectures on Finite Precision Computations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.

[17] C.-Y. Chen. *Adaptive numerische Quadratur und Kubatur mit automatischer Ergebnisverifikation*. PhD thesis, Universität Karlsruhe, 1998.

[18] J.-M. Chesneaux. *Etude théorique et implémentation en ADA de la méthode CESTAC*. PhD thesis, Université Pierre et Marie Curie, Paris, 1988.

[19] J.-M. Chesneaux. CADNA : An ADA tool for round-off error analysis and for numerical debugging. In *ADA in Aerospace*, Barcelona, Spain, December 1990.

[20] J.-M. Chesneaux. Study of the computing accuracy by using probabilistic approach. In C. Ullrich, editor, *Contribution to Computer Arithmetic and Self-Validating Numerical Methods*, pages 19–30, IMACS, New Brunswick, New Jersey, USA, 1990.

[21] J.-M. Chesneaux. CADNA : Un outil pour le contrôle des arrondis et le débogage numérique en ADA. *La lettre ADA*, 48–49:14–22, November–December 1991.

[22] J.-M. Chesneaux. Descriptif d'utilisation du logiciel CADNA_F. Technical Report 92-32, Laboratoire MASI, June 1992.

[23] J.-M. Chesneaux. The quality relations in scientific computing. *Num. Algo.*, 7:129–143, 1994.

[24] J.-M. Chesneaux. *L'arithmétique stochastique et le logiciel CADNA*. Habilitation à diriger des recherches, Université Pierre et Marie Curie, Paris, 1995.

[25] J.-M. Chesneaux, L.-S. Didier, and F. Rico. Fixed CADNA library. In *Proc. 5th Real Numbers and Computers conference*, Lyon, France, September 2003.

[26] J.-M. Chesneaux and F. Jézéquel. Dynamical control of computations using the trapezoidal and Simpson's rules. *J. of Universal Computer Science*, 4(1):2–10, 1998.

[27] J.-M. Chesneaux and J. Vignes. Sur la robustesse de la méthode CESTAC. *C. R. Acad. Sci. Paris Sér. I Math.*, 307:855–860, 1988.

[28] J.-M. Chesneaux and J. Vignes. Les fondements de l'arithmétique stochastique. *C. R. Acad. Sci. Paris Sér. I Math.*, 315:1435–1440, 1992.

[29] R. Chotin-Avot. *Architectures matérielles pour l'Arithmétique Stochastique Discrète*. PhD thesis, Université P. et M. Curie, Paris, France, June 2003.

[30] S.D. Conte and C. de Boor. *Elementary numerical analysis*. McGraw-Hill, International Student edition, 1980.

[31] R. Cools. Advances in multidimensional integration. *J. Comput. Appl. Math.*, 149:1–12, 2002.

[32] R. Cools and A. Haegemans. CUBPACK: progress report. In T. O. Espelid and A. Genz, editors, *Numerical Integration - recent developments, software and applications*, volume 357 of *NATO ASI Series C: Mathematics and Physical Sciences*, pages 1–24. Kluwer Academic, Dordrecht, 1992.

[33] R. Cools and A. Haegemans. Algorithm 824: CUBPACK: A package for automatic cubature; framework description. *ACM Trans. Math. Softw.*, 29(3):287–296, September 2003.

[34] R. Cools, D. Laurie, and L. Pluym. Algorithm 764: CUBPACK++: A C++ package for automatic two-dimensional cubature. *ACM Trans. Math. Softw.*, 23(1):1–15, March 1997.

[35] R. Cools and P. Rabinowitz. Monomial cubature rules since 'Stroud': a compilation. *J. Comput. Appl. Math.*, 48:309–326, 1993.

[36] G.F. Corliss, Ch. Hu, R. Ramirez, R.B. Kearfott, and G.W. Walster. Proposed level-1 interval basic linear algebra software (INTBLAS) subpackage for the new BLAS. Technical report, Marquette University and Sun Microsystems, 1997.

[37] M. Daumas and J.-M. Muller, editors. *Qualité des calculs sur ordinateur*. Masson, Paris, 1997.

[38] P.J. Davis and P. Rabinowitz. *Methods of numerical integration*. Academic Press, London, 1984.

[39] H. Engels. *Numerical quadrature and cubature*. Academic Press, 1980.

[40] P.C. Hammer and A.W. Wymore. Numerical evaluation of multiple integrals I. *MTAC*, 11:59–67, 1957.

[41] G.I. Hargreaves. Interval analysis in MATLAB. Numerical Analysis Report No. 416, Manchester Centre for Computational Mathematics, University of Manchester, December 2002. Available at `http://www.maths.man.ac.uk/~nareports/narep416.pdf`.

[42] N.J. Higham. *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.

[43] O. Holzmann, B. Lang, and H. Schütt. Newton's constant of gravitation and verified numerical quadrature. *Reliab. Comput.*, 2(3):229–239, 1996.

[44] M. Iri. History of automatic differentiation and rounding error estimation. In A. Griewank and G.F. Corliss, editors, *Automatic Differentiation of Algorithms: Theory, Implementation and Application*, pages 3–16. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1991.

[45] F. Jézéquel. Dynamical control of converging sequences computation. *Applied Numerical Mathematics*, 50(2):147–164, 2004.

[46] F. Jézéquel and J.-M. Chesneaux. For reliable and powerful scientific computations. In Krämer and Wolff von Gudenberg, editors, *Scientific Computing, Validated Numerics, Interval Methods*, pages 367–378. Kluwer Academic/Plenum Publishers, 2001.

[47] F. Jézéquel and J.-M. Chesneaux. Computation of an infinite integral using Romberg's method. *Num. Algo.*, 36(3):265–283, July 2004.

[48] U.W. Kulisch. *Advanced Arithmetic for the Digital Computer*. Springer-Verlag, Wien, 2002.

[49] J.-L. Lamotte. A tool for the numerical validation of parallel code programming with Fortran 77 and PVM. In *Proc. 2nd european conference EuroPVM*, pages 173–178, Lyon, France, 1995. Hermes.

[50] J.-L. Lamotte. Parallelization of the CESTAC method on shared memory and distributed memory computers. In *10th International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics, SCAN2002, Paris, France*. GAMM-IMACS, September 2002.

[51] J.-L. Lamotte and D. Martins. First parallel implementation of the CESTAC method with self validation. Mathematical Modelling and Scientific Computations, minisymposium within the 33rd Spring Conference of the Union of the Mathematicians in Bulgaria, Borovets, Bulgaria, April 2004.

[52] B. Lang. Verified quadrature in determining Newton's constant of gravitation. *J. Univ. Comput. Sci.*, 4(1):16–24, 1998.

[53] B. Lang. Derivative-based subdivision in multi-dimensional verified Gaussian quadrature. In G. Alefeld, J. Rohn, S. Rump, and T. Yamamoto, editors, *Symbolic Algebraic Methods and Verification Methods*, pages 145–152, Wien, 2001. Springer-Verlag.

[54] P. Langlois. Automatic linear correction of rounding errors. *BIT*, 41(3):515–539, September 2001.

[55] P. Langlois. *Précision finie et méthodes automatiques.* Habilitation à diriger des recherches, Université Claude Bernard Lyon 1, July 2001.

[56] P. Langlois. Improving the CENA method with the IA'64, Itanium(TM) processor. In *Proc. SCAN2002 conference, Paris, France*, September 2002.

[57] S. Linnainmaa. Towards accurate statistical estimation of rounding errors in floating-point compuations. *BIT*, 15:165–173, 1975.

[58] S. Linnainmaa. Error linearization as an effective tool for experimental analysis of the numerical stability of algorithms. *BIT*, 23:346–359, 1983.

[59] J.N. Lyness. Symmetric integration rules for hypercubes. I, II, III. *Math. Comp.*, 19:260–276, 394–407, 625–637, 1965.

[60] J.N. Lyness. An error functional expansion for N-dimensional quadrature with an integrand function singular at a point. *Math. Comp.*, 30:1–23, 1976.

[61] J.N. Lyness and R. Cools. A survey of numerical cubature over triangles. In W. Gautschi, editor, *Mathematics of computation 1943-1993: A half-century of computational mathematics, Proceedings of Symposia in Applied Mathematics*, volume 48, pages 127–150. American Mathematical Society, Providence, Rhode Island, 1994.

[62] J.N. Lyness and E. de Doncker. Quadrature error expansions. II. The full corner singularity. *Numer. Math.*, 64:355–370, 1993.

[63] J.N. Lyness and B.J.J. McHugh. On the remainder term in the N-dimensional Euler-Maclaurin expansion. *Numer. Math.*, 15:333–344, 1970.

[64] J.N. Lyness and K.K. Puri. The Euler-Maclaurin expansion for the simplex. *Mathematics of computation*, 27(122):273–293, 1973.

[65] S. Markov and R. Alt. Stochastic arithmetic: addition and multiplication by scalars. *Applied Numerical Mathematics*, 50(3–4):475–488, 2004.

[66] W. Miller. Software for roundoff analysis. *ACM Trans. Math. Softw.*, 1:108–128, 1975.

[67] M. Montagnac. *Contrôle dynamique d'algorithmes itératifs de résolution de systèmes linéaires.* PhD thesis, Université Pierre et Marie Curie, Paris, 1999.

[68] W.J. Morokoff and R.E. Caflisch. Quasi-Monte Carlo Integration. *J. Comp. Phys.*, 122(2):218–230, 1995.

[69] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods.* SIAM, Philadelphia, 1992.

[70] D.S. Parker. Monte Carlo Arithmetic: Exploiting randomness in floating-point arithmetic. Technical Report 970002, UCLA Computer Science Dept., 1997. Available at `http://www.cs.ucla.edu/~stott/mca/CSD-970002.ps.gz`.

[71] D.S. Parker, P.R. Eggert, and B. Pierce. Monte Carlo Arithmetic: A framework for the statistical analysis of roundoff error. Technical Report 970014, UCLA Computer Science Dept., 1997. Available at `http://www.cs.ucla.edu/~stott/mca/CSD-970002.ps.gz`.

[72] D.S. Parker, B. Price, and P. Eggert. Monte Carlo Arithmetic: How to gamble with floating point and win. *Computing in Science and Engineering*, pages 58–68, July–August 2000.

[73] S.M. Rump. INTLAB - INTerval LABoratory. In T. Csendes, editor, *Developments in reliable computing. SCAN-98 conference, 8th international symposium on Scientific computing, computer arithmetic and validated numerics. Budapest, Hungary, 1998*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.

[74] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[75] A. Sidi. Euler-Maclaurin expansions for integrals over triangles and squares of functions having algebraic/logarithmic singularities along an edge. *J. Approx. Theory*, 39:39–53, 1983.

[76] J. Stewart. *Calculus: Early Transcendentals*. Brooks/Cole Publishing Company, 5th edition, 2002.

[77] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*, volume 12 of *Texts in applied mathematics*. Springer, 3rd edition, 2002.

[78] U. Storck. Numerical integration in two dimensions with automatic result verification. In *[2]*, pages 187–224.

[79] M. Tienari. A statistical model of roundoff error for varying length floating-point arithmetic. *BIT*, 10:355–365, 1970.

[80] R. van der Pas and B. Walster. Interval arithmetic support in the Sun Fortran compiler. In *NSF workshop on Reliable Engineering Computing, Savannah, Georgia, USA*, September 2004.

[81] P. Verlinden and A. Haegemans. An error expansion for cubature with an integrand with homogeneous boundary singularities. *Numer. Math.*, 65:383–406, 1993.

[82] P. Verlinden, D.M. Potts, and J.N. Lyness. Error expansions for multidimensional trapezoidal rules with Sidi transformations. *Num. Algo.*, 16(3-4):321–347, 1997.

[83] J. Vignes. Zéro mathématique et zéro informatique. *C. R. Acad. Sci. Paris Sér. I Math.*, 303:997–1000, 1986. also: La Vie des Sciences, 4 (1) 1-13, 1987.

[84] J. Vignes. Estimation de la précision des résultats de logiciels numériques. *La Vie des Sciences*, 7(2):93–145, 1990.

[85] J. Vignes. A stochastic arithmetic for reliable scientific computation. *Math. Comput. Simulation*, 35:233–261, 1993.

[86] J. Vignes. A stochastic approach to the analysis of round-off error propagation. A survey of the CESTAC method. In *Proc. 2nd Real Numbers and Computers conference*, pages 233–251, Marseille, France, 1996.

[87] J. Vignes and R. Alt. An efficient stochastic method for round-off error analysis. In W.L. Miranker and R.A. Toupin, editors, *Accurate scientific computations*, volume 235 of *Lecture Notes in Computer Science*, pages 183–205, Symposium Bad Neuenahr, Germany, March 1985. Springer Verlag.

[88] J. Vignes and M. La Porte. Error analysis in computing. In *Information Processing 1974*, pages 610–614. North-Holland, 1974.

[89] S. Wedner. *Verifizierte Bestimmung singulärer Integrale - Quadratur und Kubatur*. PhD thesis, Universität Karlsruhe, 2000.

[90] J.H. Wilkinson. Rounding errors in algebraic processes. In *Notes on Applied Science*, number 32. Her Majesty's Stationery Office, London, 1963. Also published by Prentice-Hall, Englewood Cliffs, New Jersey, USA. Reprinted by Dover, New York, 1994.

[91] J.H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, 1965.

[92] J.H. Wilkinson. Modern error analysis. *SIAM review*, 13(4):548–568, 1971.

[93] J.H. Wilkinson. Error analysis revisited. *IMA Bulletin*, 22(11-12):192–200, 1986.

Chapter 5

Major articles

We present in this chapter the following documents:

- J.-M. Chesneaux, F. Jézéquel, *Dynamical control of computations using the trapezoidal and Simpson's rules*, J. of Universal Computer Science, 4(1): 2-10, 1998.
- F. Jézéquel, J.-M. Chesneaux, *For reliable and powerful scientific computations*, Scientific Computing, Validated Numerics, Interval Methods, Krämer and Wolff von Gudenberg ed., Kluwer Academic/Plenum Publishers, 367-378, 2001.
- F. Jézéquel, *Dynamical control of converging sequences computation*, Applied Numerical Mathematics, 50(2): 147-164, 2004.
- F. Jézéquel, J.-M. Chesneaux, *Computation of an infinite integral using Romberg's method*, Numerical Algorithms, 36(3): 265-283, 2004.
- F. Jézéquel, F. Rico, J.-M. Chesneaux, M. Charikhi, *Reliable computation of a multiple integral involved in the neutron star theory*, article submitted to "Mathematics and Computers in Simulation".

5.1 On the dynamical control of the trapezoidal rule and Simpson's rule

J.-M. Chesneaux, F. Jézéquel,
Dynamical control of computations using the trapezoidal and Simpson's rules,
J. of Universal Computer Science, 4(1): 2-10, 1998.

5.2 On the estimation of round-off error propagation on vector computers

F. Jézéquel, J.-M. Chesneaux,
For reliable and powerful scientific computations,
Scientific Computing, Validated Numerics, Interval Methods, Krämer and Wolff von Gudenberg ed., Kluwer Academic/Plenum Publishers, 367-378, 2001.

5.3 On the dynamical control of converging sequences computation

F. Jézéquel,
Dynamical control of converging sequences computation,
Applied Numerical Mathematics, 50(2): 147-164, 2004.

5.4 On the computation of an improper integral involved in crystallography

F. Jézéquel, J.-M. Chesneaux,
Computation of an infinite integral using Romberg's method,
Numerical Algorithms, 36(3): 265-283, 2004.

5.5 On the computation of a multiple integral involved in the neutron star theory

F. Jézéquel, F. Rico, J.-M. Chesneaux, M. Charikhi,
Reliable computation of a multiple integral involved in the neutron star theory,
article submitted to “Mathematics and Computers in Simulation”.