HENRI PESONEN
NUMERICAL INTEGRATION IN BAYESIAN POSITIONING

Master of Science Thesis

Topic accepted by the Department Council on 8.6.2005
Examiner: professor Robert Piché (TUT)
Preface

This thesis is the result of my work in positioning algorithms research group at the Department of Mathematics of Tampere University of Technology. I started my work on positioning in November 2004 as a part-time employee and continued my work full-time after spring 2005. This thesis was written during the winter 2005-06 which turned out to be an extremely educational period of time for me.

I would like to thank professor Robert Piché, who was the examiner of this thesis and whose lectures on numerical mathematics I have greatly enjoyed. I would also like to thank my co-workers: M.Sc. Simo Ali-Löytty, M.Sc. Niilo Sirola, Duane Petrovich and Tommi Perälä. Thanks goes also to my family, friends and the staff at the Department of Mathematics. Also, I would like to acknowledge the support of Nokia Corporation who have funded this research.

Tampere, 18th May 2005

Henri Pesonen
Contents

Abstract 

Tiivistelmä

Abbreviations and Acronyms

Symbols

1 Introduction

2 Bayesian Estimation in Local Positioning
   2.1 Local positioning .................................................. 14
   2.2 Bayesian estimation ............................................... 18
   2.3 Bayesian filtering ................................................. 22
   2.4 Evaluating integrals in Bayesian statistics .................... 24

3 Numerical Integration
   3.1 Fundamentals of numerical integration .............................. 26
      3.1.1 Compound integration rules ................................. 28
   3.2 Polynomial formulas .............................................. 30
      3.2.1 Formula construction using symmetry .................... 37
      3.2.2 Romberg Integration ........................................ 39
   3.3 Adaptive methods .................................................. 42
      3.3.1 Adaptive algorithms ....................................... 42
      3.3.2 Simplicial grid refinement ................................. 47
   3.4 Monte Carlo methods ............................................. 49

4 Testing and Simulations
   4.1 Compared integration methods .................................... 53
      4.1.1 Reference solutions ......................................... 55
   4.2 Positioning tests .................................................. 56
      4.2.1 2D-tests ...................................................... 59
      4.2.2 3D-tests ...................................................... 59
      4.2.3 4D-tests ...................................................... 61
5 Conclusions and Future Work

Bibliography

A Cubature Rules
  A.1 Interpolatory Cubature Rules
Abstract

Bayesian estimation theory offers a general framework for mathematical modeling that makes possible the introduction of subjective knowledge of the problem into the model in the form of the prior information. One of the drawbacks of this flexibility is the need to frequently evaluate multi-dimensional integrals numerically. One application where Bayesian approach have proven its value is positioning and tracking. The position estimate usually has to be obtained online and in an environment where resources can be very limited using often noisy measurements. The aim usually is to obtain an adequately accurate result relatively quickly because the exact answer of the mathematical model is not necessarily the exact position of the object to be located.

In this work we study different multidimensional numerical integration methods and their applicability to the positioning problem. One of the problems related to multidimensional numerical integration is the dimensionality effect, which refers to the fast growth of the computational load as the dimension of the problem grows. To alleviate this problem, different methods have been studied, including adaptive integration methods which are the focus in this thesis. Adaptive methods are tested against sampling based Monte Carlo method that is often used to evaluate multi-dimensional integrals. In testing section, we generated two-, three- and four-dimensional integrals that would occur when Bayesian framework is applied to local positioning. Results show that adaptive method excels in two- and three-dimensional tests, but the dimensionality effect favors Monte Carlo methods in four-dimensional tests.

Bayesianinen teoria on todistanut käyttökelpoisuutensa monissa tekniiikan alueissa kuten esimerkiksi paikannuksessa ja seurannassa. Bayesianen lähestymistapa mahdollistaa kaikenlaisten mitatausten hyväksikäytön matemaattisessa mallissa sekä soveltuu rekursiivisessa muodossaan erinomaisesti seurantaan.

Eri paikannusmenetelmistä erittäin suosituksi on noussut GPS-paikannus (Global Positioning System), joka perustuu satelliittien lähetettiin signaaloihin. GPS-signaalille voidaan käyttää tarkan paikan ratkaisemiseen varsinkin tilanteissa, joissa useisiin satelliitteihin on suora näkyvyys. Sisätiloissa ja kaupunkialueilla satelliittipaikannuksen tarkkuus ja luotettavuus kuitenkin heikkenevät heijastusten sekä näköasteiden vuoksi. Satelliittipaikannusta tukenaan näissä tilanteissa on tutkittu lokaaleja paikannusmenetelmiä, joita voitaisiin käyttää myös omina menetelminä.

Lokaali paikannus on sovellus, jossa paikkamestaattia selvitetään käyttäen paikallisaa signaalialaa esimerkiksi Bluetooth-, WLAN- sekä matkapuhelintukiasemistassa. Henkilökohtaisella paikannuksella lokaaleissa ympäristöissä on valtavasti erilaisia...
käyttömahdollisuuksia, ja sillä onkin hyvin suuret markkinanäkymät. Matkapuhelimitä soitzetun häätäpuheluiden paikantaminen on sovellus, joka on todettu erittäin hyödylliseksi. Lisäksi logistisia sovelluksia yrityksille ja esimerkiksi sairaaloille on jo nyt olemassa. Tulevaisuudessa sovelluksia tulee olemaan varmasti monenlaisia.


Koska lokaalissa paikanmuoksessa integroitjualue voidaan usein rajata monitahokkaaksi, tai sitä voidaan hyvin approksimoida monitahokkaalla, on tässä työssä valittu perusintergrointialueeksi simpleksi. Monitahokas voidaan aina ja kaa simpleksien yhdistelmäksi. Monitahokas voidaan automaattisesti muokata useisiin tapoihin, ja se on ylellinen haitan integroitoukseen. Yksi moniulotteiseen numeeriseen integroitiiin liittyvä ongelma on niin sanottu dimensionalisaatiokirou, joka viittaa nopeasti kasvavaan laskennallisuuden vaatimuuteen kun integroatointgelman dimensio kasvaa. Tämän ongelman vaikutusten lieventämisessä on tutkittu useita eri menetelmiä, mutta integroitulla on adaptiivisuuteen ja satunnaisuuteen perustuvia menetelmiä. Satunnaisuuteen perustuvien menetelmien kuten Monte Carlo -integraalin etuna on suppenneminen riippumattomuus integroatointigelman dimensiosta sekä yksinkertainen implementointi. Hattapuolelta on menetelmän hidas suppenneminen.

Työn testausosiossa muodostamme testi-integraaleja erilaisista paikanmuokskaarioistoista. Tässä ongelmassa esiintyyliä integraaleilla, jotka ovat todennäköisyystiheysfunktiointi, on usein yhteistä todennäköisyysmassan keskiyminen verrattain pienelle osalle integroatointialueesta. Testi-integraalit olivataksi-, kolmi- tai nelulotteisia. Referenssiapikaestimaattina käytettiin apikaestimaattia, joka oli perusteltu adaptiivisella menetelmällä jakamalla integroatointialue erittäin monen simpleksin yhdistelmäksi. Testattavina menetelmänä olivat kolme eri adaptiivista menetelmää sekä Monte Carlo -menetelmä. Testitulokset osoittavat, että odotetusti
adaptiiviset menetelmät antavat huomattavasti tarkempia tuloksia kuin Monte Carlo -menetelmä kaksi- ja kolmiulotteisissa testeissä, mutta neliulotteisissa testeissä dimensionaalisuuden kirous suosii jo selvästi Monte Carloa.
## Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>Third Generation</td>
</tr>
<tr>
<td>AGPS</td>
<td>Assisted Global Positioning System</td>
</tr>
<tr>
<td>AOA</td>
<td>Angle-of-arrival</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>Cell-ID</td>
<td>Cell identity</td>
</tr>
<tr>
<td>DOD</td>
<td>The United States Department of Defense</td>
</tr>
<tr>
<td>EKF</td>
<td>Extended Kalman Filter</td>
</tr>
<tr>
<td>Eq.</td>
<td>Equation</td>
</tr>
<tr>
<td>FCC</td>
<td>The United States Federal Communications Commission</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>MATLAB</td>
<td>Matrix Laboratory, a mathematical software</td>
</tr>
<tr>
<td>MS</td>
<td>Mobile Station</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>RSS</td>
<td>Received Signal Strength</td>
</tr>
<tr>
<td>RTD</td>
<td>Round Trip Delay</td>
</tr>
<tr>
<td>RTT</td>
<td>Round Trip Time</td>
</tr>
<tr>
<td>TA</td>
<td>Timing Advance</td>
</tr>
<tr>
<td>TDOA</td>
<td>Time Difference of Arrival</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunication System</td>
</tr>
<tr>
<td>UTC</td>
<td>Universal Coordinated Time</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local-Area Network</td>
</tr>
</tbody>
</table>
Symbols

\[ \int \quad \text{integral} \]
\[ \int_B \quad \text{integral over region } B \]
\[ \approx \quad \text{approximate equality} \]
\[ \propto \quad \text{proportionality} \]
\[ \sim \quad \text{asymptotic equality} \]
\[ := \quad \text{equality by definition} \]
\[ \rightarrow \quad \text{convergence} \]
\[ \lfloor x \rfloor \quad \text{largest integer less than or equal to } x \]
\[ |x| \quad \text{absolute value of } x \]
\[ \| \cdot \| \quad \text{norm} \]
\[ \| \cdot \|_\infty \quad \text{max norm} \]
\[ \| \cdot \|_1 \quad \text{L}_1 \text{ norm} \]
\[ \infty \quad \text{infinity} \]
\[ \wedge \quad \text{logical and} \]
\[ \lor \quad \text{logical or} \]
\[ \cup \quad \text{union} \]
\[ \cap \quad \text{intersection} \]
\[ \binom{i}{j} \quad \text{binomial coefficient} \]
\[ A^{-1} \quad \text{inverse of matrix } A \]
\[ A^T \quad \text{transpose of matrix } A \]
\[ \text{argmax}_f(\cdot) \quad \text{argument that maximizes } f(\cdot) \]
\[ B_n(\cdot) \quad \text{Bernoulli polynomial} \]
\[ \tilde{B}_n(\cdot) \quad \text{periodic Bernoulli polynomial with unit length} \]
\[ BS_i \quad \text{i}^{\text{th}} \text{ base station} \]
\[ c \quad \text{speed of light} \]
\[ c_i \quad \text{cubature rule weight} \]
\[ \mathbb{C} \quad \text{set of complex numbers} \]
\[ \mathbb{C}^N \quad \text{set of } N\text{-times continuously differentiable functions} \]
\[ \mathbb{C}_N^+ \quad \text{N-dimensional positive unit cube } [0, 1]^N \]
\[ \text{Cov}[f, g] \quad \text{covariance of } f \text{ and } g \]
\[ \frac{\partial}{\partial x} \quad \text{partial differential operator} \]
\[ \deg P \quad \text{degree of polynomial } P \]
det \quad \text{determinant}
\dim(d, N) \quad \text{dimension of space of } N\text{-variate polynomials of maximum degree } d
\mathbf{d} \quad \text{multi index}
\epsilon \quad \text{requested error}
E(f; B) \quad \text{quadrature or cubature error operator on } f \text{ over } B
\hat{E}_{p,n}(f; B) \quad \text{random quadrature or cubature error operator on } f \text{ over } B
E[x] \quad \text{expectation of random variable } x
E[x; y = y] \quad \text{expectation of random variable } x \text{ with condition } y = y
\eta \quad \text{difficulty parameter}
\exp \quad \text{exponential function}
erf(x) \quad \text{error function}
\varepsilon \quad \text{random measurement error}
\bar{f}(x) \quad \text{average of } f \text{ evaluated at points } G(x)
\mathcal{F} \quad \text{sigma algebra of event space}
F(\cdot) \quad \text{affine transformation}
f_k(\cdot) \quad \text{motion function}
G \quad \text{linear transformation}
G_s f \quad \text{Grundmann-Möller integration rule of degree } 2s + 1 \text{ over } T_N
\mathcal{G} \quad \text{set of linear transformations}
\mathcal{G}(x) \quad G\text{-orbit of a point } x
\chi_B(\cdot) \quad \text{characteristic function of } B
H(\cdot) \quad \text{Heaviside's function}
h_k(\cdot) \quad \text{measurement function}
I_w, I \quad \text{integral operator}
\inf \quad \text{infimum}
J_h \quad \text{Jacobian of } h
(k \times Q) \quad \text{copy rule}
\lim \quad \text{limit}
l(x; y) \quad \text{likelihood function}
\log \quad \text{logarithm}
\text{max} \quad \text{maximum}
\mu \quad \text{expectation value}
\mathbb{N} \quad \text{set of natural numbers}
n! \quad \text{factorial}
\mathcal{N}(\mu, \sigma^2) \quad \text{normal distribution with mean } \mu \text{ and variance } \sigma^2
O(\cdot) \quad \text{Landau symbol}
\Omega \quad \text{event space}
\omega \quad \text{event}
\prod \quad \text{product}
P(\cdot) \quad \text{probability measure}
p(x) \quad \text{probability density function}
p_x(x) \quad \text{probability density function of } x
SYMBOLS

$p(x, y)$  joint probability density function
$p_{x,y}(x, y)$  joint probability density function of $x$ and $y$
$p(x|y)$  conditional probability density function
$p_{x|y}(x|y)$  conditional probability density function of $x$ with condition that $y = y$
$\mathcal{P}_D^N$  space of $N$-variate polynomials of maximum degree $d$
$\mathcal{P}_D^N(\mathcal{G})$  Space of $N$-variate polynomials of maximum degree $d$
invariant with respect to $\mathcal{G}$
$\emptyset$  empty set
$\phi_b(\cdot)$  radical-inverse function in the base $b$
$\pi$  permutation
$Qnf, Q(f; B)$  quadrature or cubature operator on $f$ (over $B$)
$\hat{Q}_{p,n}f$  random cubature rule with $n$ abscissas sampled from $p$ on $f$
$(Q_{n_1} \times Q_{n_2})f$  product rule cubature operator on $f$
$R^{\mu,\alpha}B$  $\mu$–panel offset trapezoidal rule operator over region $B$
$r_i$  measurement to $i^{th}$ base station
$\mathbb{R}$  set of real numbers
$S_N$  symmetric group
$\sum$  sum
$\sum f(x)_S$  sum over all distinct permutations of $x$
$\sum f((x))_S$  sum over all distinct permutations of $N + 1$-vector $x = (1 - \sum_{i=1}^{N} x_i, x_1, \ldots, x_N)^T$
$\sigma$  standard deviation
$\sigma(x)$  standard deviation of random variable $x$
$\hat{\sigma}$  sample mean of standard deviation
$\text{supp}(f)$  support of $f$
$t_i$  time measurement at position $i$
$T_N$  $N$-dimensional simplex
$T_{x_0,k,\pi}$  subsimplex defined by a vertex $x_{0,k}$ and permutation $\pi$
$v_k$  noise in the measurement model
$\text{Var}[f]$  variance of $f$
$\text{vol}(B)$  volume of a region $B$
$w_k$  noise in the dynamic model
$w(x)$  weight function
$\{x_k\}_k$  sequence $x_1, x_2, \ldots$
$\xi(x)$  the number of different subsimplices into which $x \in T_N$ belongs after $T_N$ has been subdivided
$\mathbf{Y}_{1:k}$  $\{y_i, i \in 1, 2, \ldots, k\}$
$\zeta$  parameter determining the location of the peak of a Gaussian testfunction
Chapter 1

Introduction

Numerical approximation of integrals has been an area of mathematics that has been of great interest for hundreds, even thousands of years. At first this meant determination of areas and volumes of physical objects but today computational integration is applied in areas like theoretical physics and computer graphics. Many great scientists have studied this art including Newton, Kepler, Maxwell and even Archimedes who obtained strict bounds for \( \pi \) by approximating a circle with inscribed and circumscribed polygons [Engels 1980]. However, it is perhaps the contribution of Gauss that has influenced the modern numerical integration the most. He noticed that by placing the evaluation points in some other way than equidistantly, more accurate results can be obtained. Nowadays numerical integration is still much studied area of applied mathematics. Like many areas of numerical mathematics it still keeps evolving as more and more sophisticated algorithms are being developed and as computing technology is advancing rapidly, old techniques once considered to be computationally too intensive might be applicable today.

One branch of mathematics that generate analytically unsolvable integration problems is statistics, and particularly Bayesian statistics which is named after reverend Thomas Bayes (1702-1761) (biography by [Bellhouse 2004]). As well as being an excellent tool for mathematical modeling, Bayesian approach leaves us with high-dimensional integration problems that naturally demand lot of computing power. This is actually
why Bayesian estimation was first thought to be unsuitable for practical problems. We introduce the basic concepts of Bayesian estimation theory in Section 2.2.

One of the many technologies where Bayesian estimation theory has proven its value is positioning and tracking. Determining the location of an object and keeping track of it is important for military and industrial purposes, but during the last decade or so, personal positioning for consumers has become a problem of great interest. Personal positioning and its applications are considered to have great commercial potential which is helped by the fact that the prices of GPS-receivers have dropped considerably and the receiver units are much more compact than they used to be. Based on signals from satellites, GPS provides a good position estimate in many cases but in urban environments and indoors GPS has difficulties caused by reflections and refractions of the signals. The Bayesian approach allows the fusion of all kinds of measurements and data, so we could assist GPS with more local measurements when this is necessary. This is one reason for studying positioning using signals from nearby sources such as Bluetooth, WLAN and cellular networks. Concepts of local positioning are presented in Section 2.1.

This work concentrates on approximate integration in local positioning. Because measurements in this environment are relatively noisy or in some cases even completely wrong and evaluation has to be done online, low number of function evaluations is considered more important than normally, even at the expense of accuracy. In Chapter 3 we discuss numerical integration methods in detail. We present some fundamentals of numerical integration in Section 3.1 and in Section 3.2 we go over some details about interpolatory integration formulas. Sampling based Monte Carlo integration methods are discussed in Section 3.4. To alleviate the fast computational growth as the dimension of the problem grows we study adaptive algorithms in Section 3.3.1. Throughout this work, the basic integration region is considered to be N-dimensional simplex which is a more versatile building block than the traditional hypercube.

In the numerical testing part Chapter 4 we generate test integrals that resemble the integration problems that arise in local positioning. We compare sampling based Monte Carlo integration method with interpolatory integration methods in two-, three- and four-dimensional test cases. In Chapter 5 we draw conclusions from the test results and provide guidelines for future research.
Chapter 2

Bayesian Estimation in Local Positioning

In this chapter, we discuss local positioning and tracking, and particularly the Bayesian framework for representing uncertainties involved in the process.

First, in Section 2.1, we introduce the concept of local positioning and mention some techniques and measurements used to carry out the positioning. The rest of the chapter is devoted to the Bayesian estimation theory and its application to the positioning problem. In Sections 2.2 and 2.3, we first present the concepts of Bayesian estimation and filtering on a general level and then how they are used in positioning. In the final Section 2.4, we present the problem that the methods discussed in this work are used to solve.

2.1 Local positioning

Wireless data transfer technology has advanced greatly during the last few years. As a result, this technology has been integrated into devices such as mobile phones, laptop computers and personal digital assistants that use these wireless signals for various purposes. It is of great interest to exploit these signals in as many ways as
possible. One task is to install location awareness feature into these devices. Location awareness means that somehow the device is aware of its surroundings and, if possible, is able to interact with nearby objects. In addition to location awareness, location estimation and target tracking are other problems that can be solved using wireless signals. Of course, there is a vast amount of interest in commercial applications for private personnel in the area of positioning. Also logistic applications are much in demand as companies want locate their resources and rescue workers benefit from even approximately knowing where the emergency call came from. New ideas for using positioning surface frequently. A few of them can be found listed for example, in Sayed et al. [2005]. From location-aware advertising to interactive maps, a number of different kinds of applications are under development or already have been developed.

Global Positioning System (GPS) positioning gives reliable estimate of the location but might not work as reliably in cities or indoors. It is one reason for the research of using signals from nearby sources such as cellular networks or wireless local area networks networks (WLANs) to aid GPS or to be used as a stand-alone positioning methods. This location estimation in a more compact environment is referred to as local positioning. It contains all positioning technology that is used to locate a mobile station (MS) with respect to one or more devices with known positions. The implementations of positioning methods can be divided into three categories: network-based, MS-assisted network-based and MS-based network-assisted methods, depending where the major part of the computation is done [Syrrjärinne 2001].

In network-based implementation all the necessary measurements and computing is done in the network independent of MS whereas in MS-assisted network-based implementation the mobile station is involved in the positioning process. MS-based network-assisted methods use mobile station to do most of the measurements and computing and network is used only to provide assistance to the process. It depends on the situation which implementation is the most feasible one, since a wrong one could raise the costs considerably [Vossiek et al. 2003].

Measurement principles

Principal radio signal based positioning techniques use triangulation to determine the position. Triangulation include lateration and angulation, which use distance and angle measurements respectively. The most used distance measurement principles are Received-Signal-Strength (RSS), Time-Difference-of-Arrival (TDOA), Round-Trip-Delay (RTD) and Time-of-Arrival (TOA). Angle measurement is usually referred to as Angle-of-Arrival (AOA) [Vossiek et al. 2003].

AOA is based on the direction where the signals are coming from. Using for example an array of antennas or directional antennas, the angle between some fixed direction and the direction where the signal is coming from can be determined. Since AOA
gives a straight line from the MS to an antenna the location of the device can be
determined with multiple AOA measurements from different antennas by calculating
the intersection of the lines.

With RSS measurements, the distance of the MS from an antenna can be determined
since the free-space transmission attenuation of the signal is proportional to $1/r^2$,
where $r$ is the distance from a transmitter to a receiver. Propagation-time based
measurements like TOA, TDOA and RTD are used to calculate the position by using
time differences between detection of the signals in different sensors. To do this, the
positions of the sensors and the velocity of the signal are needed. TOA is based on
measuring the time that a signal took to travel from transmitter to a receiver. This
gives the distance $r_i$ between the MS and the base station $BS_i$

$$r_i = (t_i - t_{MS})c,$$  \hspace{1cm} (2.1)

where $c$ the speed of light, $t_{MS}$ is the time of transmission from MS and $t_i$ is the time
when $BS_i$ received the signal. In a plane, to obtain an unambiguous position, distance
measurements to at least three different locations are needed [Hightower and Borriello
2001], [Sayed et al. 2005].

A major technical difficulty with TOA based measurements is that they need accu-
rately synchronized clocks in the system, including the mobile station. Global System
for Mobile Communications (GSM) and Third Generation (3G) cellular networks
are not synchronized. To use TDOA measurements, a synchronized network is not
necessary since measurable values are time differences between the arrival of signal in
different receivers,

$$r_{ij} = r_i - r_j = (t_i - t_{MS})c - (t_j - t_{MS})c = (t_i - t_j)c,$$  \hspace{1cm} (2.2)

in which $t_{MS}$ does not appear.

In RTD based positioning, the measured time is the round-trip time that the signal
takes to travel from the transmitter to a destination and back to the transmitter.

$$r_i = \frac{1}{2}((t_i - t_{MS})c + (t_{MS_2} - t_i)c) = \frac{1}{2}(t_{MS_2} - t_{MS_1})c.$$  \hspace{1cm} (2.3)

Using RTD measurements all that is required is an accurate clock integrated with the
transmitters.

Since positioning using the measurement principles mentioned above is based on trans-
mitted signals and received signals, there are many factors that affect the quality of
the signal. Reflections and refractions of the signal from objects cause an effect called
multipath [Kaplan 1996], in which the receiver detects the signal from multiple direc-
tions at different times, so that the true direction and other data contained in the
signal is compromised.

In addition to the radio signal based measurements, there exist various other measure-
ments that can be used to complement the positioning. These include measurements
using different sensors such as accelerometers, gyroscopes, compasses and barometers.[Syrjärinne 2001]

Positioning with cellular networks

Positioning cellular phones has received a considerable amount of research during the last years [Drane et al. 1998], [Syrjärinne 2001]. One factor behind this interest is the implementation of Enhanced 911 (E911) for wireless services [FCC 1996]. In 1996 the United States Federal Communications Commission (FCC) adopted a report and order that mobile service providers would have to provide a location of a 911 caller. Similar requirements are planned in Europe also.

Using existing base station (BS) network in positioning is of course cost effective and is under considerable amount of research. Using proximity analysis to provide a location estimate is probably the most natural positioning method since the network is divided into cells each served by its own base station and each cell has its own identity (Cell-ID) and if a mobile phone is served by a particular base station we know that we are inside its coverage area. Problem is that this method alone might not provide a position accurate enough since cell radius can be as large as 35km or even larger with extended cells [Syrjärinne 2001]. This performance can be upgraded as more than one base station can be located at the same area, each covering a different region. A common configuration is to use three base stations in one site each covering a 120° sector of a cell [Drane et al. 1998].

Other measurements that are currently being used in mobile phone positioning are for example TOA, RSS, TDOA, AOA and RTD, which is referred to as timing advance (TA) in GSM networks and as round trip time (RTT) in 3G networks. Since the existing cellular networks were planned for communication purposes and not for positioning purposes, the accuracy of measurements is not sufficient for certain purposes. For instance, TA and RTT measurements are quantized with resolutions $\approx 550m$ and $\approx 80m$ respectively. This means that the measurement errors are of about the same accuracy and this does not include the degrading propagation effects [Syrjärinne 2001]. Even though the mentioned measurements are far from optimal, the possibility to combine Cell-ID with range measurements can be used to provide a crude position estimate fast and the availability and applicability are excellent [Borkowski et al. 2004].

Positioning with wireless local area networks and Bluetooth

Similar to cellular networks, WLANs and Bluetooth can be used for positioning purposes. With the popularity of wireless access to Internet and wireless local area networks it seems reasonable to research the use of 802.11 WLANs for various different
purposes, including positioning. The most natural way of positioning in a wireless network is using proximity analysis. Because it is relatively simple for the device to access the network via the access point that gives the best signal, we can use the known location of the access point as our rough location estimate. Since the coverage area of an access point is limited, approximately 23m in diameter [Tanz and Shaffer 2005], this might be sufficient for some purposes. Triangulation is also used with success. In Bahl and Padamanabhan [2000] the authors obtained accuracy of about 4.3m for the 50th percentile. But since WLAN position is carried out indoors, the wave propagation can be rather complicated in an environment that has many walls and moving objects.

Bluetooth signals [Bluetooth 2001] are also being integrated in increasingly many wireless devices. Bluetooth signals can be used in positioning similarly as WLANs [Kotanen et al. 2003] and the easiest way to obtain a position estimate is to use information about the nearest access points. A typical power-class 3 Bluetooth device has a maximum range of 10m, so in this case the theoretical worst case error is 10m. Also Bluetooth positioning gains from using triangulation as reported for example by Hallberg et al. [2002].

**Assisted Global Positioning System**

The most used method of positioning and certainly the most covered one, has been to use GPS-signals. The Global Positioning System was designed by the Department of Defense of United States for military purposes, but it is also freely available for public use. GPS is based on 24 satellites orbiting Earth at an altitude of over 20000 km in 4 different orbital planes. This means that with a clear view of the sky, there are always 6-12 satellites visible and we can get very accurate position estimates. TOA measurements are used to determine position, which is possible since satellites utilize an internal GPS system time and the clocks of the receivers are synchronized with Coordinated Universal Time (UTC).

One problem with GPS is that a clear view of the sky is often not available in urban environment and indoors. GPS measurements can be combined with measurements from nearby sources to help positioning when compromised performance of GPS is detected. This combined method is referred to as assisted GPS (AGPS), which currently is considered to be one of the best methods for cellular positioning [Borkowski et al. 2005].

### 2.2 Bayesian estimation

Many engineering applications have to deal with a problem of determining unknown variables of interest by inferring information from other observable variables. Observ-
variables are usually outcomes of experiments or measurements. This is a problem of mathematical modeling that might be solvable with many different approaches, depending on the situation. One procedure that especially has gained popularity during the last decades is the Bayesian approach. In the Bayesian approach we take all the unknown parameters in the problem to be random. We obtain information about unknown variables of interest by developing probabilistic models for data variables we can observe and using our prior knowledge of the unknown variables.

In this work, it is assumed that readers are familiar with the basic properties of probability theory used in the following sections. Therefore we are only going to introduce some concepts and notation that are used throughout the rest of the chapter. For a more in-depth review of the basic principles of statistics and probability theory, [Shiryayev 1984] and [Williams 2001] are suggested sources.

The probability space is denoted by a triple $(\Omega, \mathcal{F}, P)$, where $\Omega$ is the event space, $\mathcal{F}$ is a sigma algebra of $\Omega$ and $P$ is a probability measure in $\Omega$. Event $\omega \in \Omega$ is an elementary event. A random variable is a $\mathcal{F}$-measurable function $\Omega \rightarrow \mathbb{R}$. We write random variables with bold face font such as $\mathbf{y}$ and refer to its realized value $\mathbf{y}(\omega)$ as $y$. No notational separation is being made between a random variable and a random vector, which is a vector of random variables. The probability density function of random variable $\mathbf{y}$ of some event space $\Omega$ is marked formally by $p_\mathbf{y}(\mathbf{y})$, but a shorter notation $p(y)$ is used in places where it is unambiguous. The joint probability density function of two random variables is denoted as $p(x, y)$ or $p_{x,y}(x, y)$. Probability density function $p(x|y = y)$, that is the probability of $x$ with condition that $\mathbf{y}$ has realized as $y$, is marked as $p(x|y)$ or $p_{x|y}(x \mid y)$. The mean value of $\mathbf{x}$ is denoted as $E[\mathbf{x}]$ and $E[x|y]$ refers to the mean value of $\mathbf{x}$ with condition $\mathbf{y} = y$.

**Definition 2.1 (Bayes’ rule).** Conditional probability density function of random variable $\mathbf{x}$ with condition $\mathbf{y} = y$ is defined by the equation, when $p(y) > 0$

$$p(x|y) = \frac{p(x, y)}{p(y)}. \quad (2.4)$$

Bayes’ rule gives the probability of $x$ with the condition that $\mathbf{y}$ has realized as $y$. This formula is the basis for Bayesian approach to statistics and in fact one could argue that it is the only formula in Bayesian statistics.

Now we use the fact that

$$p(y) = \int p(y|x)p(x)dx$$

and by applying the Bayes’ rule again we can write (2.4) in a form from which we can observe how the conditional probabilities are related to each other when $p(y) \neq 0$

$$p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)dx}. \quad (2.5)$$
The problem of determining (2.5) is a fundamental one in Bayesian statistics as often we want to say something about \( x \) given \( y \). We introduce some common terminology that we use in the following text. The conditional probability density function \( p(x|y) \) is the \textit{posterior probability distribution function}. The variable \( x \) is usually a state vector, and \( y \) is a data vector such as a vector of measurements that give us some information about \( x \). To determine the posterior probability distribution we need the \textit{prior probability distribution function} \( p(x) \) and the \textit{sampling distribution function} \( p(y|x) \).

There is also another way of thinking the sampling distribution \( p(y|x) \). If we fix \( y = y_0 \) and think of \( p(y_0|x) \) as a function of \( x \) we obtain something that is referred to as a \textit{likelihood function}

\[
l(x; y_0) := p(y_0|x).
\]  

(2.6)

Likelihood function is chances that for any legal value of \( x \), we have observed \( y_0 \) [Gamerman 1997]. It is important to notice that the likelihood function does not have to be a probability distribution. In other words, integrating it over the whole space is not necessarily one. We incorporate the likelihood function into (2.5) and obtain the form

\[
p(x|y) = \frac{l(x; y)p(x)}{\int l(x; y)p(x)dx} \propto l(x; y)p(x).
\]

(2.7)

We have the posterior distribution \( p(x|y) \) and we are interested in \( x \). We need to have some kind of estimate for location as \( p(x|y) \) is a rather complex estimate of the position. Because the posterior distribution contains all the information of the problem, we can obtain an optimal estimate, by any criteria, from it. Two such estimates are the maximum a posteriori state, which is the maximum point of the posterior, and the mean value of the posterior distribution [Roos et al. 2002]. Finding the global maximum point of the posterior distribution is generally a nonlinear programming problem that can be extremely difficult to solve when the posterior distribution has multiple peaks. Finding the mean of the posterior usually requires solving a high dimensional integration problem which will be discussed in Section 2.4.

Bayesian estimation in local positioning

In a deterministic scenario, the position estimate could be obtained by computing the intersection of the loci of the measurements [Sirola et al. 2003]. One of the problems is that inaccurate measurements might be incompatible. The statistical approach does not have a problem with incompatible measurements as all measurements, including unlikely ones are still possible with some probability.

When applying Bayesian estimation to a positioning or tracking problem, we want to obtain the distribution of the parameter \( x \) which contains the location of the mobile station and possibly other variables of interest such as velocity. The observation
parameter vector $y$ might contain a variety of different measurements such as the ones described in Section 2.1. The randomness in the observation vector $y$ can be thought to be the random noise in measurements or an effect of imprecise sensors, that unfortunately always exists in the real world. Usually if we do not have any other knowledge of the location of a device than that we are inside some bounded region, we take a uniform density as our prior distribution.

Next, consider the positioning example with range measurements in the Bayesian framework. We denote the unknown location to be determined with $x$ and use $r_i$ to denote the range measurement from the base station $i$ with known position $s_i$. Then, a range measurement can be formulated as

$$ r_i = \|s_i - x\| + \epsilon_i, \quad (2.8) $$

where $\epsilon_i$ denotes the error in the measurement that is taken to be a random variable with probability density $p_{\epsilon_i}$. Using this notation we can formulate the measurement likelihood function as

$$ p(r_i|x) = p_{\epsilon_i}(r_i - \|s_i - x\|). \quad (2.9) $$

Sector information and maximum range could be treated as a measurement and be expressed as

$$ p(r_i|x) = \chi_B(x) = \begin{cases} 1 & x \in B \\ 0 & x \notin B \end{cases}, \quad (2.10) $$

where $B$ is the region bounded by maximum range and sector information. Range measurement with Gaussian noise and sector information is illustrated in Figure 2.1.

---

**Figure 2.1:** The measurement likelihood function using range measurement with Gaussian noise and Cell-Id with sector information
If we have obtained several measurements \( r_1, \ldots, r_n \) that have independent errors, we can define \( r := (r_1, \ldots, r_n)^T \) and formulate the measurement likelihood as

\[
p(r|x) = \prod_{i=1}^{n} p(r_i|x).
\] (2.11)

A situation when two range measurements is obtained and sector information is used is illustrated in Figure 2.2.

![Figure 2.2: Combination of all available measurements](image)

**2.3 Bayesian filtering**

In positioning and tracking, the position of the receiver has to be computed on-line as more measurements are obtained. The Bayesian framework is suitable for doing this because Bayesian estimation can be used recursively by updating the model with new incoming measurements or observations. The positioning problem is then to recursively determine the posterior \( p(x_k|y_{1:k}) \) of the state \( x_k \) with the set of all realized measurements \( y_{1:k} = \{y_n, n = 1, \ldots, k\} \). To improve the model by using the prior information, we need a new model to describe how the state evolves with time in addition to the measurement model.

**Definition 2.2 (Stochastic Process).** Let \((\Omega, \mathcal{F}, P)\) be a probability space and \(T\) a set of parameters. A stochastic process is a mapping \( x : \Omega \times T \to \mathbb{R}^N \) such that with every \( t \in T \), \( x(\cdot, t) \) is a measurable mapping \( \Omega \to \mathbb{R}^N \).

**Definition 2.3 (Markov process of order one).** A discrete-time stochastic process is a Markov process of order one if

\[
p(x_{k+1} = j|x_k = i, x_{k-1} = i_{k-1}, \ldots, x_0 = i_0) = p(x_{k+1} = j|x_k = i)
\] (2.12)

with every \( k \) and \( i_{k-1}, \ldots, i_0 \).
**Definition 2.4** (White Noise). A stochastic process is said to be white noise if it is a Markov process and
\[ p(x_{k+1}|x_k) = p(x_{k+1}) \]  
(2.13)

To define the state and the measurement models, we consider the state and the observation vectors to be stochastic processes \( \{x_k, k \in \mathbb{N}\}, \{y_k, k \in \mathbb{N}\} \) respectively. We formulate the dynamics of the system as stochastic difference equations
\[ x_k = f_k(x_{k-1}, w_{k-1}), \]  
(2.14)

where \( f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_x} \), and for the measurements
\[ y_k = h_k(x_k, v_k), \]  
(2.15)

where \( h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_y} \). We need to have the distribution of \( x_0 \) as the initial condition. Random variables \( w_k \) and \( v_k \) stand for the noises in the dynamics and the measurements respectively. They are assumed to be white noise and independent of each other. Both the state transition function \( f_k \) and the measurement function \( h_k \) are possibly nonlinear functions.

The recursive part is divided into two parts prediction and update, with initial step when we have no measurements, \( p(x_0) \) considered to be known. Also, the statistics of \( w_k \) and \( v_k \) are considered to be known. The state model (2.14) is taken to be a Markov process of order one, so that the current state depends only on the state of the previous time step. Then the predictive part is the Chapman-Kolmogorov equation
\[ p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}, \]  
(2.16)

where \( p(x_k|x_{k-1}) \) represents how the state evolves and depends on Eq. (2.14). Applying Bayes’ rule, the update part is formulated as
\[ p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}, \]  
(2.17)

where the normalizing constant is
\[ p(y_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k. \]

This recursive process can unfortunately be solved analytically only in a few very restrictive special cases. The Kalman filter [Kalman 1960] solves the filtering problem optimally if the prior and the noises in dynamics and measurements are Gaussian and (2.14) and (2.15) are linear functions. Modifying the Kalman filter algorithm to perform local linearization of \( f_k \) and \( h_k \) and to approximate (2.17) as Gaussian we obtain the Extended Kalman filter (EKF) [Grewal and Andrews 1993]. Bayesian filters for positioning and tracking are discussed in numerous articles, for example in...
ones by Fox et al. [2003] and Arulampalam et al. [2002]. Grid-based methods [Bucy and Senne 1971], [Kramer and Sorenson 1988] solve the filtering problem optimally in the case of discrete state space consisting of a finite number of states. Otherwise they can be used to approximately solve the problem. Grid-based methods rely strongly on numerical integration methods which are discussed in Chapter 3.

2.4 Evaluating integrals in Bayesian statistics

As it has been observed, Bayesian statistics frequently involve integrals that have to be evaluated in some way. For example, the marginal density

$$\int l(x; y)p(x)dx$$

which is the reciprocal of the normalizing constant of the posterior distribution, needs to be evaluated. Also, if the posterior mean is taken to represent the posterior distribution, integrals of the form

$$E[x|y] = \int xp(x|y)dx = \frac{\int xp(y|x)p(x)dx}{\int p(y|x)p(x)dx}$$ \hspace{1cm} (2.18)

need to be evaluated. This integral is often analytically intractable and often relatively high dimensional as the integral has as many dimensions as $x$. The integral of the vector $xp(x|y)$ is taken componentwise. All of the integrals in the process have to be evaluated at every time-step of the process and considering the fact that the measurements can be very noisy, we need to consider which is more important: very good approximation of $E[x|y]$ obtained after intensive computation or a quickly obtained adequate approximate of it.

Because it is almost always impossible to deal with equations of the form (2.18) analytically, different strategies have been developed to solve it approximately. These strategies include restriction of models to a few simple ones, analytic approximations, and numerical approximations. Numerical methods include sampling-based methods and interpolatory integration methods. These methods will be discussed in detail in Chapter 3 and their performance will be compared in Chapter 4. The performance of different integration methods in positioning problem has also been tested elsewhere, for example in Sirola et al. [2005].

When evaluating integrals, we have to keep in mind the positioning and tracking problem that is the recursive problem defined by equations (2.16) and (2.17). To have a good approximation for this problem, we need to consider how the different integration methods work when applied in the recursive problem.
Chapter 3

Numerical Integration

This chapter considers methods for solving integration problems numerically. The focus is on multidimensional integration. The research on numerical integration methods has been ongoing for a long time, and well-known elegant methods are many, at least for one dimensional problems. But there doesn’t exist a method that would be perfect for all different problems with different characteristics. So the research continues as the fast development in computer technology ensures that more and more computing power is becoming available. Computing power is one reason why as well studied as the problem one-dimensional numerical integration is, there are still advances made in the field of multi-dimensional numerical integration.

The structure of this chapter is as follows: Section 3.1 gives some fundamentals of numerical integration and notations we are using and Section 3.2 covers interpolatory integration formulas. We discuss much-studied subjects like formula construction (3.2.1), extrapolation (3.2.2) and compound formulas (3.1.1). More thorough discussion can be found for example in Cools [1997], Davis and Rabinowitz [1984] and Krommer and Ueberhuber [1998]. Section 3.3 considers methods that use different ways to adapt themselves to integration problems at hand, that is adaptive integration methods and algorithms (3.3.1) and some discussion about simplicial grids (3.3.2) is also included with an algorithm for subdividing a \( N \)-dimensional simplex. In Section
3.4 we discuss sampling based integration methods and in particular, we go over the basic theory of Monte Carlo integration.

### 3.1 Fundamentals of numerical integration

In the following we refer to numerical approximation of integrals in one dimension as *quadrature* and in higher dimensions *cubature*.

The problem considered in this and in the following sections is, how to evaluate the integral of a function \( f : A \to \mathbb{R} \) over the \( N \)-dimensional region \( B \subset A \subset \mathbb{R}^N \) with respect to a weight function \( w : \mathbb{R}^N \supset B \to \mathbb{R} \)

\[
I_w f := \int_B w(x)f(x)dx,
\]

(3.1)

The notation \( I f \) is used whenever \( w(x) = 1 \).

Analytically this problem would be solved using iteration of integrals to divide the problem into multiple one-dimensional integrals and applying the fundamental theorem of integral calculus, but for most of the time we are unfortunately unable to do so. This might be because integrand is too difficult, or even impossible to express in closed form, which is common in the case of multidimensional integration. Or it could be that we don’t even know function \( f \) analytically, only as function values evaluated at some points in space, or maybe we know it only as a computer program or we obtain the function through a computer procedure. When the analytical solution is unobtainable, we need a numerical approximation of the integral.

The integral operator is a linear functional operator and so should be quadrature and cubature operators. All quadrature and cubature formulas are weighted sums. Usually they can be written as

\[
Q_n(f;B) := \sum_{k=1}^n c_k g_k(x_k) \approx I_w f = \int_B w(x)f(x)dx.
\]

(3.2)

where \( c_k \in \mathbb{R} \) are weights of cubature formula and \( g_k \) are some functions. Usually \( g_k = g = f \), but it is important to notice that it could be, for example, an approximation of \( f \) that is easier to handle but preserves some distinct features of \( f \), also some cubature formulas use \( g_k = f' \). Also, it is important to note \( \{x_1, \ldots, x_n\} \) does not have to be a subset of \( B \). In this work, the points \( x_k \) in which cubature formulas evaluate the functions are referred to as abscissas. When the integration region does not have to be specified, we write

\[
Q_n f := Q_n(f;B)
\]

The operator used to approximate the error of the numerical integration is denoted by

\[
E(f;B) \approx |Q_n(f;B) - I_w f|.
\]

(3.3)
What kind of properties would we like a cubature formula to have? It is a desirable property that a method integrates exactly functions of some particular space. For example, many formulas are exact for functions of a space of certain degree of algebraic or trigonometric polynomials. In addition, it should give good approximations when used to integrate functions that do not belong to that space. A method should be efficient so that it would use as few function evaluations as possible to get a desired result. Unfortunately, we almost always have to balance between accuracy and efficiency. Cubature formula should use only abscissas that are located inside the integration region since obtaining function values outside integration region in practical applications does not always make sense. Cubature formula should preferably use only positive $w_i$. This is because a rule with negative weights can possibly give a negative integral for a positive function and numerical stability might become an issue since cubature rules consist of sums of many terms.

The curse of dimensionality

A major problem of interpolatory integration methods (discussed in the next section) is an effect called the curse of dimensionality or the dimensionality effect. It refers to the fast growth of computational complexity as the dimension of the integration problem grows. A simple example by Davis and Rabinowitz [1984] illustrates this effect. Let the entire N-dimensional space be divided into a equally spaced cubical grid with a side length $h$, where $h$ is small. Then let the volume of the N-dimensional integration region $B$ be denoted as $\text{vol}(B)$ and let $m$ mark the number of cubes contained entirely in $B$. Then we know that

$$\text{vol}(B) = \int_B \text{d}x \sim m h^N. \quad (3.4)$$

The error made in this estimation is approximately the volume of cubes that are only partly in $B$. If we denote the error of the approximation as

$$E(f; B) = |I f - Q_n(f; B)| \sim h S, \quad (3.5)$$

where $S \sim \frac{\text{vol}(B)}{\text{vol}(B)^{1/N}}$ is the surface measure of $B$. Combining (3.4) and (3.5) we get

$$\frac{E(f; B)}{\text{vol}(B)} \sim \frac{1}{m^{1/N}}.$$ 

This shows that as the dimension of the problem grows the number of points needed to obtain a fixed relative error estimate grows exponentially. Although interpolatory integration always suffers from the dimensionality effect, different schemes such as adaptive algorithms (discussed in Section 3.3) have been applied successfully to alleviate the effects of the problem.
3.1.1 Compound integration rules

Cubature formulas are almost always constructed for certain standard regions that appear often in integration problems. These $N$-dimensional regions include cube, ball, sphere and simplex

$$T_N := \left\{ (x_0, \ldots, x_N) : \sum_{i=0}^{N} x_i \leq 1, x_i \geq 1 \right\}. \quad (3.6)$$

In this work we construct cubature formulas for simplex.

Using the additivity property of integrals

$$\int_B f(x) \, dx = \int_{B_1} f(x) \, dx + \int_{B_2} f(x) \, dx + \cdots + \int_{B_n} f(x) \, dx = \sum_{i=1}^{n} \int_{B_i} f(x) \, dx, \quad (3.7)$$

and the multivariate transformation rule, we can construct cubature rules for all regions that can be divided into standard ones.

**Definition 3.1.** An affine transformation is a mapping $F : \mathbb{R}^n \to \mathbb{R}^n$

$$F(x) = Ax + b, \quad (3.8)$$

where $A \in \mathbb{R}^{n \times n}$ is a non-singular transformation matrix and $b \in \mathbb{R}^n$ is a translation vector.

**Theorem 3.2** (Multivariate Transformation Rule). Let $f : B \to \mathbb{R}$ be a Riemann integrable function and let $h : \mathbb{R}^n \supset A \to B$ be a continuously differentiable function such that its Jacobian matrix $J_h$ is nonsingular on $A$, that is $\det J_h \neq 0$, $\forall x \in A$. Then

$$\int_B f(u) \, du = \int_A f(h(x)) |\det J_h(x)| \, dx \quad (3.9)$$

**Proof.** See Rudin [1976] page 252. \qed

Often we have a case when we have a cubature formula constructed for a standard region such as $T_N$ and we want to integrate over another simplex. Then we can use an affine transformation $h(x) = F(x)$. This procedure is relatively simple because of the simple form of the determinant $\det J_F = \det A$.

Similar to the multivariate transform rule, we can transform cubature formula for different regions.

Let $h(x) = u$ be a continuously differentiable linear function such that the determinant of its Jacobian matrix $\det J_h \neq 0$, $\forall x \in A$ and

$$I_w f \approx Q_n f = \sum_{i=1}^{n} c_i f(u_i)$$
Using Theorem 3.2 we can express the cubature formula constructed for region $B$ as a cubature formula for region $A$

$$\int_{A} f(h(x)) \left| \det J_h(x) \right| dx \approx \sum_{i=1}^{n} c_i \left| \det J_h(x) \right| f(h(x_i)) \quad (3.10)$$

Let the integration region $B$ be partitioned into regions $B := B_1 \cup \ldots \cup B_n$ whose interiors are disjoint. This can be used to construct compound integration rules. If we have a cubature rule for each region $B_l$ such that

$$Q_{m_l}(f; B_l) := \sum_{i=1}^{m_l} c^{(l)}_i f(x^{(l)}_i), \; l = 1, \ldots, n,$$

they can be combined as

$$I_w f = \int_B w(x)f(x)dx = \int_{B_1} w(x)f(x)dx + \ldots + \int_{B_n} w(x)f(x)dx \approx \sum_{l=1}^{n} Q_{m_l}(f; B_l) = \sum_{l=1}^{n} \sum_{i=1}^{m_l} c^{(l)}_i f(x^{(l)}_i).$$

The error for the compound integration formula can be obtained via

$$E(f; B) = \sum_{l=1}^{n} Q_{m_l}(f; B_l) - I_w f = \sum_{l=1}^{n} Q_{m_l}(f; B_l) - \left( \int_{B_1} w(x)f(x)dx + \ldots + \int_{B_n} w(x)f(x)dx \right) = \sum_{l=1}^{n} \left( Q_{m_l}(f; B_l) - \int_{B_l} w(x)f(x)dx \right) = \sum_{l=1}^{n} E_l(f; B_l).$$

If the integration region $B$ is divided into $k$ integration regions of the same type and size and the same integration rule $Q_n$ is applied in each of the regions, then this special case of compound rule is called a copy rule and is denoted as $(k \times Q_n)$. Let $Q_{n_l} := (k_l \times Q_n)$ and $\{Q_{n_1}, Q_{n_2}, \ldots \}$ be a sequence of copy rules such that $k_l \leq k_{l+1}$. Using acceleration algorithms [Krommer and Ueberhuber 1998] we can speed up the convergence of a sequence $\lim_{l \to \infty} Q_{n_l} f = I_w f$. In Section 3.2.2, we discuss the Romberg integration, which is one of the most used methods of accelerating convergence.

The number of abscissas in the compound cubature formula can generally be expected to be $nm_l$, or $kn$ in the case of copy rules. But if some of the abscissas lie on the boundary of regions $B_l$ then the same abscissas might belong to several subregions.
and the amount of function evaluations can reduce considerably. This situation is illustrated in Figure 3.1, in which we have a cubature formula for a polygonal region that use the midpoints of edges of the subregions as abscissas and a rule which uses abscissas that lie inside the subregions (the derivation of these formulas can be found in Section 3.2.1). Note that with fine grids with large amount of subregions, this effect is emphasized and in higher dimensions even more so.

![Figure 3.1: Abscissas on the boundaries decrease the amount of distinct abscissas](image)

3.2 Polynomial formulas

One desirable property of a cubature formula is that it is exact for functions of a particular function space. This gives us some perspective of how well a given formula performs with an integrand at hand. Most commonly used approximation spaces are \( N \)-dimensional spaces of algebraic and trigonometric polynomials of degree \( d \). Only spaces of algebraic polynomials are considered in this work. Polynomials are extensively used to interpolate and approximate functions and first we are going to review some of their properties.

**Definition 3.3** (Degree of Multivariate Polynomials). The degree of a multivariate monomial \( x^d = x_1^{d_1}x_2^{d_2} \cdots x_N^{d_N} \) is \( \| d \|_1 = |d_1| + |d_2| + \cdots + |d_N| \). Because any polynomial can be expressed as a linear combination of monomials as \( P = \sum_{j=1}^J c_j x^{d_j} \), the degree of a nonzero polynomial is the degree of its monomial that has the highest degree.

\[ \deg P = \max_i \{ \| d_i \|_1 : i = 1, \ldots, J \}. \]  

[Krommer and Ueberhuber 1998]

**Theorem 3.4.** The dimension of the \( N \)-variate polynomial space of polynomials of maximum degree \( d \), \( \mathcal{P}_d^N = \text{span}\{x^d : d \in \mathbb{N}^N, \| d \|_1 \leq d\} \) is

\[ \dim(d, N) = \binom{N + d}{N} \]  

(3.11)
Proof. In Engels [1980] page 239 can be found a proof that there are \( \binom{N + d - 1}{d} \) linearly independent \( N \)-variate polynomials of degree \( d \) that are orthogonal to the \( N \)-variate polynomials of degree \( d - 1 \). The dimension of \( \mathcal{P}_d^N \) is then

\[
\sum_{i=0}^{d} \binom{N + i - 1}{i} \tag{3.12}
\]

We need to show that Eq. (3.12) = Eq. (3.11), which we do using induction.

\[d = 0 : \sum_{i=0}^{0} \binom{N + i - 1}{i} = \binom{N - 1}{0} = 1 = \binom{N + 0}{N}.
\]

\[d = k - 1 : \text{Assume true}
\]

\[d = k : \sum_{i=0}^{k} \binom{N + i - 1}{i} = \sum_{i=0}^{k-1} \binom{N + i - 1}{i} + \binom{N + k - 1}{k}
\]

\[= \binom{N + k - 1}{N} + \binom{N + k - 1}{k}
\]

\[= \frac{(N + k - 1)!}{N!(k - 1)!} + \frac{(N + k - 1)!}{k!(N - 1)!}
\]

\[= \frac{(N + k)(N + k - 1)!}{N!k!} = \binom{N + k}{N}
\]

The fact that we are able to integrate polynomials in closed form over most standard regions makes them an excellent tool for constructing cubature formulas and corresponding error estimation procedures. The most common way of constructing rules is to demand that the rule is exact for a certain polynomial space.

**Definition 3.5.** A cubature formula is said to be of degree \( d \) if

\[
\forall f(x) \in \mathcal{P}_d^N \Rightarrow Q(f; B) = If \quad \text{and} \quad \exists q \in \mathcal{P}_{d+1}^N \quad \text{such that} \quad Q(q; B) \neq Iq
\]

Since almost always cubature formulas are derived for standard regions like the unit cube, we have to change the given integration region. It can be proved that the algebraic degree of the cubature formula does not change under this affine transformation.

**Theorem 3.6.** The algebraic degree of cubature formula does not change under affine transformation.


Although we do not know how our cubature formulas performs when it is applied to a function that is not of algebraic degree of \( d \) or less, the following Theorem gives confidence that higher degree formulas gives better approximations than formulas of lower degree.
Theorem 3.7. For all cubature formulas $Q_n$ of accuracy $d \geq 0$ that have only positive weights $c_i > 0$
\[ |Q_n f - I_w f| \leq \left( \| w \|_1 + \sum_{i=1}^{n} |c_i| \right) e^*_d(f), \]  
(3.13)

where
\[ e^*_d(f) := \inf \left\{ \| P_d - f \|_\infty : P_d \in \mathcal{P}_d \right\} \]

Proof. In Krommer and Ueberhuber [1998] the proof has been provided for the one-dimensional case. The multi-dimensional case is analogous:

Let $P \in \mathcal{P}_d^N$. Then $Q_n P = I_w P \Leftrightarrow Q_n P - I_w P = 0.$

\[ |Q_n f - I_w f| = |Q_n f - Q_n P + I_w P - I_w f| \leq |Q_n f - Q_n P| + |I_w P - I_w f| = |Q_n (f - P)| + |I_w (P - f)| \]

Then the upper bounds can be obtained

\[ |Q_n (f - P)| = \left| \sum_{i=1}^{n} c_i (f(x_i) - P(x_i)) \right| \leq \sum_{i=1}^{n} |c_i| |(f(x_i) - P(x_i))| \leq \max_{i=1,...,n} |f(x_i) - P(x_i)| \sum_{i=1}^{n} |c_i| \leq \max_{x \in \mathbb{R}^N} |f(x) - P(x)| \sum_{i=1}^{n} |c_i| \]

\[ |I_w (P - f)| = \left| \int_B w(x)(P(x) - f(x))dx \right| \leq \int_B |w(x)||((P(x) - f(x))| dx \leq \max_{x \in \mathbb{R}^N} |P(x) - f(x)| \int_B |w(x)| dx \]

Although Theorem 3.7 can not be directly used to compute an upper bound for the error, we can expect to get better results by using cubature formulas of higher degree because $e^*_d \leq e^*_{d-1}$. Also, if the integrand is a smooth function so that it can be well approximated with a polynomial, formulas with arbitrary weights of higher degree could be expected to give better results than formulas of lower degree, because higher degree formulas are exact for a larger space of functions.
If we want to derive a cubature formula that is exact for polynomials of degree \( d \) or less, the formula must be such that
\[
Q_n(f; B) = I f, \quad \forall f \in \mathcal{P}^N_d.
\] (3.14)

To fulfill Eq. (3.14), it is necessary and sufficient that the formula is able to integrate exactly some basis \( \{b_1, \ldots, b_{\text{dim}(d,N)}\} \) of the space \( \mathcal{P}^N_d \), because
\[
f \in \mathcal{P}^N_d \Rightarrow f = \sum_{i=1}^{\text{dim}(d,N)} a_i b_i
\]
\[
\Rightarrow \int f(x) dx = \int \sum_{i=1}^{\text{dim}(d,N)} a_i b_i(x) dx = \sum_{i=1}^{\text{dim}(d,N)} a_i \int b_i(x) dx.
\]

We formulate a set of equations called the moment equations:
\[
Q_n b_j = \sum_{i=1}^n c_i b_j(x_i) = I b_j, \quad j = 1, \ldots, \text{dim}(d,N).
\] (3.15)

If we have a set of prescribed abscissas \( X = \{x_1, \ldots, x_n\} \) then by solving the linear set of equations (3.15) we obtain the weights \( c_i \) of the cubature formula. As an example we derive a cubature formula exact for polynomials of degree \( d \leq 2 \) for the three-dimensional simplex \( T_3 \). The basis functions of the space \( \mathcal{P}^3_2 \) are taken to be the monomials
\[
\{1, x_1, x_2, x_3, x_1 x_2, x_1 x_3, x_2 x_3, x_1^2, x_2^2, x_3^2\}.
\]

We choose the prescribed abscissas as
\[
X = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/2 \\ 0 \\ 1/2 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1/4 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/4 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1/4 \end{bmatrix} \right\},
\]
which is illustrated in Figure 3.2. Then the set of moment equations expressed in matrix form is
\[
\begin{bmatrix}
1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\
0 \ 1 \ 0 \ 0 \ 1/2 \ 0 \ 0 \ 1/2 \\
0 \ 0 \ 1 \ 0 \ 0 \ 1/2 \ 0 \ 1/2 \\
0 \ 0 \ 0 \ 0 \ 0 \ 1/2 \ 0 \ 0 \\
0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1/4 \ 0 \\
0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1/4 \\
0 \ 1 \ 0 \ 0 \ 1/4 \ 0 \ 0 \ 1/4 \\
0 \ 0 \ 1 \ 0 \ 0 \ 1/4 \ 0 \ 0 \\
0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1/4 \ 0 \\
0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1/4 \ 0
\end{bmatrix}
\cdot
\begin{bmatrix}
c_1 \\
c_2 \\
c_3 \\
c_4 \\
c_5 \\
c_6 \\
c_7 \\
c_8 \\
c_9 \\
c_{10}
\end{bmatrix}
= \int_{T_3} \begin{bmatrix} 1 \\
x_1 \\
x_2 \\
x_3 \\
x_1 x_2 \\
x_1 x_3 \\
x_2 x_3 \\
x_1^2 \\
x_2^2 \\
x_3^2
\end{bmatrix} dx.
CHAPTER 3. NUMERICAL INTEGRATION

Figure 3.2: Abscissas of a cubature rule exact for polynomials of degree $d \leq 2$ over $T_3$

Note that prescribed abscissas should be chosen such that the set of moment equations would not be an overdetermined system. Another way to approach the formula construction is to take abscissas to be unknown. When abscissas are unknown we have a set of nonlinear equations. Some methods that are used to solve this problem will be explored further in Section 3.2.1.

**Definition 3.8** (Interpolatory Cubature Formula). Cubature formula is called interpolatory if the weights of a formula of degree $d$ uniquely solve the system of moment equations at a given set of abscissas.

Since it is always desirable to use as few function evaluations as possible to obtain as high accuracy as possible in the algebraic sense, we would like to know what is the optimal number of points to get a certain algebraic accuracy.

**Theorem 3.9.** Let $I_w$ be an integral operator over region $B \subset \mathbb{R}^N$. The number of abscissas $n$ in any interpolatory cubature formula of degree $d$ satisfies

$$
\left( \frac{N + \lfloor d/2 \rfloor}{N} \right) \leq n \leq \left( \frac{N + d}{N} \right)
$$

(3.16)

Optimal cubature formulas that use $n$ abscissas that correspond to the lower bound in (3.16) are called minimal cubature formulas.

Unfortunately, even with minimal cubature formulas, the computational complexity of (3.15) grows rapidly with growing $N$ and $d$ as seen from (3.11). This is why we need schemes like adaptive integration or we need to take into consideration other factors such as the high symmetry of certain standard integration regions such as simplex and cube. Formula construction using symmetry is discussed in Section 3.2.1.

Product rules

The theory of quadrature is considerably more developed than the theory of cubature formulas and there exists a large amount of sophisticated quadrature formulas [Kythe and Schäferkotter 2005]. Therefore it is not surprising that quadrature formulas have been used to construct cubature formulas. This can be done via product rules.

**Definition 3.10.** Let a region $B$ be a cartesian product $B := B_1 \times B_2$ of two regions $B_1, B_2 \subseteq \mathbb{R}^n$. Let $Q_{n_1}^{(1)} f_1 = \sum_{i_1=1}^{n_1} c_{i_1}^{(1)} f_1(x_{i_1}^{(1)})$ and $Q_{n_2}^{(2)} f_2 = \sum_{i_2=1}^{n_2} c_{i_2}^{(1)} f_2(x_{i_2}^{(2)})$ be cubature (or quadrature) formulas for integrals

$$\int_{B_1} f_1(x)dx \text{ and } \int_{B_2} f_2(x)dx$$

respectively. Then a cubature formula for the integral over $B$ is

$$(Q_{n_1}^{(1)} \times Q_{n_2}^{(2)}) f = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} c_{i_1}^{(1)} c_{i_2}^{(2)} f(x_{i_1}^{(1)}, x_{i_2}^{(2)})$$

(3.17)

The product integration rules are generated for rectangles, which is quite restrictive. Therefore to use product integration over different regions we use generalized product integration rules. For example, we have an integration region that can be represented iteratively as

$$I f = \int_B f(x)dx = \int_{B_1} \int_{B_2(x^{(1)})} f(x^{(1)}, x^{(2)}) dx^{(2)} dx^{(1)}.$$  

(3.18)

We can express the cubature formula for (3.18), when there exist a continuously differentiable function $h : \bar{B}_2 \rightarrow B_2(x^{(1)})$, as

$$Q_{n_1,n_2} f := \sum_{i_1}^{n_1} \sum_{i_2}^{n_2} c_{i_1}^{(1)} c_{i_2}^{(2)} \det J_{h_{x_{i_1}^{(2)}}}(x_{i_2}^{(2)}) \ f(x_{i_1}^{(1)}, h_{x_{i_2}^{(2)}}(x_{i_2}^{(2)})),$$

(3.19)
where

$$I(f; B_1) \approx Q_{n_1}(f; B_1) = \sum_{i_1=1}^{n_1} c^{(1)}_{i_1} f_1 x^{(1)}_{i_1}$$  \hspace{1cm} (3.20)$$

$$I(f; \bar{B}_2) \approx Q_{n_2}(f; \bar{B}_2) = \sum_{i_2=1}^{n_2} c^{(2)}_{i_2} f_2 x^{(2)}_{i_2}.$$  \hspace{1cm} (3.21)

[Krommer and Ueberhuber 1998]

**Error estimation**

Polynomial cubature rules offer a straightforward way to obtain an error estimate if we would assume that a higher degree rule gives a better approximation of the integral than a lower degree rule. Let $Q_{n_1}$ denote a cubature rule of degree $d_1$ and $Q_{n_2}$ a rule of degree $d_2$ and assume that $d_1 < d_2$. In addition we assume that

$$|Q_{n_2} f - I f| \ll |Q_{n_1} f - I f|.$$  \hspace{1cm} (3.22)

Because $Q_{n_2}$ is assumed to give a more accurate approximation, we would like to use it as our approximation and use the less accurate rule $Q_{n_1}$ only to help in error estimation. If, for example, we would have

$$|Q_{n_2} f - I f| \leq \frac{1}{\alpha + 1} |Q_{n_1} f - I f|,$$  \hspace{1cm} (3.23)

where $\alpha \geq 1$, then

$$|Q_{n_2} f - I f| \leq \frac{1}{\alpha + 1} |Q_{n_1} f - I f| = \frac{1}{\alpha + 1} |Q_{n_1} f - Q_{n_2} f + Q_{n_2} f - I f| \\
\leq \frac{1}{\alpha + 1} |Q_{n_1} f - Q_{n_2} f| + \frac{1}{\alpha + 1} |Q_{n_2} f - I f|,$$

from which it follows directly that

$$\alpha |Q_{n_2} f - I f| \leq |Q_{n_1} f - Q_{n_2} f| \hspace{1cm} \Leftrightarrow \hspace{1cm} |Q_{n_2} f - I f| \leq |Q_{n_1} f - Q_{n_2} f|.$$  \hspace{1cm} (3.24)

The resulting inequality gives us an upper bound for the error if the condition (3.23) holds. Even if it does not hold, (3.24) is still often taken to be an error estimate for cubature rule $Q_{n_2}$. Extensive testing is needed for each different pair of rules $Q_{n_2}$ and $Q_{n_1}$ to make sure that (3.24) is a suitable error estimate. Computationally it is preferable to use such a pair of cubature rules that the lower degree rule uses a subset of the abscissas of the higher degree rule so that no extra function evaluations would have to be made to estimate the error. Rules fulfilling this criteria are referred to as **embedded formulas**.
CHAPTER 3. NUMERICAL INTEGRATION

3.2.1 Formula construction using symmetry

The solution of the set of moment equations (3.15) gives us the weights for the cubature formula. But because we are not able to always solve (3.15) with a set of prescribed abscissas, we must introduce new unknowns into (3.15). In other words, we take the set of abscissas \(x_1, \ldots, x_n\) to be unknown. Because every \(x_i\) has \(N\) elements, we are actually introducing \(nN\) new variables into the system. The resulting system is a set of nonlinear equations with \(\dim(d, N)\) equations and \(n + nN\) unknowns. Exploiting the symmetry of regions can be used to reduce the complexity of the system of moment equations.

Definition 3.11. A set \(\Omega\) is invariant with respect to a group \(\mathcal{G}\) if for all transformations of a group, \(G \in \mathcal{G}\), \(G(\Omega) = \Omega\). A function \(f\) invariant with respect to \(\mathcal{G}\) if \(f(x) = f(G(x))\) for all \(G \in \mathcal{G}\). Integral \(I_w\) over region \(B\) is invariant with respect to \(\mathcal{G}\) if \(w\) and \(B\) are invariant with respect to \(\mathcal{G}\). [Cools 1997]

Definition 3.12. A \(\mathcal{G}\)-orbit of a point \(x \in \mathbb{R}^N\) is a set \(\mathcal{G}(x) := \{G(x) : G \in \mathcal{G}\}\).

Using \(\mathcal{G}\)-orbits we can divide the set of abscissas of a cubature formula into generators. A subset \(\{x_{i_1}, \ldots, x_{i_K}\} \subset \{x_1, \ldots, x_n\}\) is called a set of generators [Krommer and Ueberhuber 1998] if and only if

\[\mathcal{G}(x_{i_j}) \cap \mathcal{G}(x_{i_k}) = \emptyset, \text{ for } j \neq k\]

and

\[\mathcal{G}(x_{i_1}) \cup \ldots \cup \mathcal{G}(x_{i_K}) = \{x_1, \ldots, x_n\}\]

Definition 3.13. A cubature formula is invariant with respect to a group \(\mathcal{G}\) if the weight function is invariant with respect to a group \(\mathcal{G}\), its set of abscissas is a union of \(\mathcal{G}\)-orbits and all the abscissas of the same orbit have the same weights.

In other words cubature formula can be written in terms of generators

\[Qf = \sum_{j=1}^{K} c_j \overline{f}(x_{i_j}), \quad (3.25)\]

where

\[\overline{f}(x) = \frac{1}{v} \sum_{y \in \mathcal{G}(x)} f(y),\]

in which \(v\) is the number of different points in \(\mathcal{G}(x)\).

Theorem 3.14. Let \(F\) be a vector space of functions defined on \(B\) and \(\mathcal{G}\) be a group of linear transformations acting on \(F\). Then every \(\mathcal{G}\)-invariant linear functional on \(F\) is determined by its restriction to \(F(\mathcal{G})\). [Cools 1997].

Another formulation of this result is the following, known as Sobolev’s theorem

**Theorem 3.15. (Sobolev’s theorem)** If an integral operator \( I_w \) and cubature formula \( Q_n \) are invariant with respect to \( \mathcal{G} \) then

\[
Q_n f = I_w f, \quad \forall f \in P_d^N,
\]

if and only if

\[
Q_n f = I_w f, \quad \forall f \in P_d^N(\mathcal{G}),
\]

where \( P_d^N(\mathcal{G}) \) is a set of invariant polynomials of degree at most \( d \) with respect to \( \mathcal{G} \).

It has to be noted that although the use of symmetry does not remove the need to solve a set of nonlinear equations, it can in the best case simplify the problem significantly as seen from Sobolev’s theorem. To illustrate, we derive a cubature formula exact for second order polynomials in 2 variables using the symmetry of a simplex. Another example can be found in Stoyanova [2005].

We use barycentric coordinates in the following for simplification. The barycentric coordinates of a point in \( T_2 \) are \((x_0, x_1, x_2)\), where \( x_0 = 1 - \sum_{i=1}^{2} x_i \). We look for a rule that would be invariant under coordinate interchange. This means that we are looking for a formula of the form

\[
Q_3(f; T_2) = c (f(a_1, a_2, a_2) + f(a_2, a_1, a_2) + f(a_2, a_2, a_1)) \tag{3.26}
\]

A \( d \) degree polynomial in barycentric coordinates is expressed as

\[
P_d(x) = x_0^{d_0} x_1^{d_1} x_2^{d_2}, \tag{3.27}
\]

where \( d \) is a multi-index and \( d \leq d \). We are looking for a formula that is exact for at most second degree polynomials, which are

\[
\{ x_0^0 x_1^0 x_2^0, x_1^1 x_2^0 x_3^0, x_0^1 x_2^1 x_3^0, x_1^2 x_2^2 x_3^0, x_0^0 x_1^0 x_2^1, x_1^1 x_2^1 x_3^1, x_0^1 x_1^1 x_2^0, x_1^2 x_2^2 x_3^1 \}
\]

Because of the symmetry, it is sufficient that the rule be exact for the polynomials \( x_0^0 x_1^0 x_2^0, x_1^1 x_2^0 x_3^0, x_2^2 x_2^0 x_3^0 \) and \( x_1^1 x_2^1 x_3^0 \) as the addition of the rest of the polynomials does not result any new distinct equations.
\[
\begin{align*}
&c(1+1+1) = \frac{1}{2} \\
&c(a_1^0a_2^0 + a_2^0a_1^0 + a_1^0a_2^0) = \frac{1}{6} \\
&c(a_1^2a_2^0 + a_2^2a_1^0 + a_2^0a_1^2) = \frac{1}{12} \\
&c(a_1^2a_2^0 + a_2^1a_1^0 + a_2^0a_1^2) = \frac{1}{24}
\end{align*}
\]

\[\Rightarrow \begin{cases}
  c = \frac{1}{6} \\
  c(a_1 + 2a_2) = \frac{1}{6} \\
  c(a_1^2 + 2a_2^2) = \frac{1}{12} \\
  c(2a_1a_2 + a_2^2) = \frac{1}{24}
\end{cases}\]

\[\Rightarrow \begin{cases}
  a_1 + 2a_2 = 1 \Rightarrow a_1 = 1 - 2a_2 \\
  a_1^2 + 2a_2^2 = \frac{1}{2} \Rightarrow a_2 = \frac{1}{2} \text{ or } a_2 = \frac{1}{6} \\
  2a_1a_2 + a_2^2 = \frac{1}{4}
\end{cases}\]

\[\Rightarrow (a_1 = 0 \land a_2 = \frac{1}{2}) \lor (a_1 = \frac{2}{3} \land a_2 = \frac{1}{6})\]

To express the resulted formula in $T_2$ in standard coordinates, we leave the first element off so that the abscissas of the formula are

\[
\begin{align*}
  x_1 &= \left(\frac{1}{2}, \frac{1}{2}\right)^T \\
  x_2 &= \left(0, \frac{1}{2}\right)^T \\
  x_3 &= \left(\frac{1}{2}, 0\right)^T
\end{align*}
\]

or

\[
\begin{align*}
  x_1 &= \left(\frac{1}{6}, \frac{1}{6}\right)^T \\
  x_2 &= \left(\frac{2}{3}, \frac{1}{6}\right)^T \\
  x_3 &= \left(\frac{1}{6}, \frac{2}{3}\right)^T
\end{align*}
\]

with weight $c = \frac{1}{6}$.

### 3.2.2 Romberg Integration

Romberg integration can be used to construct powerful cubature formulas of high degree using simple trapezoidal rules of low degree. The theory of Romberg integration relies on asymptotic expansions of the error functional. We follow the theory presented in articles by Lyness [1978a], Lyness [1978b], Lyness and Puri [1973] and Doncker [1979], in which Euler-Maclaurin expansions for $N$-simplex are derived.

As introduced in Lyness and Puri [1973] we take the basic trapezoidal rule to be a $\mu$-panel offset trapezoidal rule.

**Definition 3.16.** The $\mu$-panel offset trapezoidal rule operator is defined as

\[
R^{[\mu, \alpha]}(a, b)f = \frac{1}{\mu} \sum_{j=-\infty}^{\infty} [H(b - x_j) - H(a - x_j)]f(x_j), \tag{3.28}
\]

where...
CHAPTER 3. NUMERICAL INTEGRATION

\[ \mu \geq 0, \]
\[ 0 \leq a \leq b \leq 1, \]
\[ -1 \leq \alpha \leq 1, \]
\[ x_j = j - 1 + t_{\alpha}^\mu, \]
\[ t_{\alpha} = \frac{1 + \alpha}{2}, \]

and \( H \) is the Heaviside's function

\[ H(x) = \begin{cases} 
1, & x > 0 \\
\frac{1}{2}, & x = 0 \\
0, & x < 0 
\end{cases} \] (3.29)

In the next Theorem we need Bernoulli polynomials which are defined by the generating function [Abramowitz and Stegun 1964]

\[ \frac{te^{xt}}{e^t - 1} = \sum_{n=0}^{\infty} B_n(x) \frac{t^n}{n!}. \] (3.30)

Periodic Bernoulli polynomials \( \bar{B}_q(t) \) with unit period are defined as in Lyness [1978a]

\[ \bar{B}_q(x) = B_q(x), \ 0 < x < 1 \]
\[ \bar{B}_q(0) = \frac{1}{2}(B_q(0) + B_q(1)), \]
\[ \bar{B}_q(x + 1) = \bar{B}_q(x). \]

**Theorem 3.17.** Let \( f: [0, 1] \to \mathbb{R} \) be \( D + 1 \) times continuously differentiable on \([a, b] \), then the error functional representation of the \( \mu \)-panel offset trapezoidal rule operator \( R^{[\mu, \alpha]}[a, b] \) is

\[ R^{[\mu, \alpha]}[a, b] f - I f = \sum_{q=1}^{D+1} \frac{1}{\mu^q} \left[ \bar{B}_q(t_{\alpha} - \mu b) f^{(q-1)}(b) - \bar{B}_q(t_{\alpha} - \mu a) f^{(q-1)}(a) \right] \]
\[ - \frac{1}{\mu^{D+1}} \int_a^b f^{(D+1)}(t) \bar{B}_{D+1}(t_{\alpha} - \mu t) \frac{dt}{(D + 1)!}, \] (3.31)

**Proof.**

For \( \mu = \text{integer} \), see Lyness [1978a].

For \( \mu \neq \text{integer} \), see Doncker [1979].

Romberg extrapolation can be applied to a sequence \( \{ R^{[\mu_k, \alpha]}[0, 1] f \}_k = \{ R^{[\mu_0, \alpha]}[0, 1] f, R^{[\mu_1, \alpha]}[0, 1] f, R^{[\mu_2, \alpha]}[0, 1] f, \ldots \} \) where \( (\mu_k - \mu_0) \in \mathbb{N} \). Then the Romberg recursion formula is

\[ \begin{cases} 
T_0^k = R^{[\mu_k, \alpha]}[0, 1] f \\
T_p^k = T_{p-1}^{k+1} + \mu_{k,p} (T_{p-1}^{k+1} - T_{p-1}^k), 
\end{cases} \] (3.31)
where \( \mu_{k,p} = \frac{\mu^2}{\mu_{k+p} - \mu_k} \) in the case when \( R^{[\mu,\alpha]}[0,1]f \) has an expansion in even powers of \( \mu^{-1} \). This happens when \( \mu \) is an integer or 2\( \mu \) is an integer and \( t_\alpha = 0 \) or \( t_\alpha = \frac{1}{2} \). [Doncker 1979]

Let \( \{ R^{[\mu,\alpha]}[0,1]f \}_k \) have an even expansion. Then \( T^k_p \) is accurate for polynomials of degree at most

\[
\begin{cases} 
2p + 1, & \text{if } \{ \mu_k \}_k \in \mathbb{N} \\
2p, & \text{if } \{ 2\mu_k \}_k \in \mathbb{N}
\end{cases}
\]

Similar results can be obtained for simplices by generalizing the theory of one dimensional case.

**Definition 3.18.** The \( N \)-dimensional simplex product trapezoidal operator is defined as a product of iterated one dimensional operators

\[
R^{[\mu,\alpha]}T_N f = R^{[\mu,\alpha_1]}[0,1]R^{[\mu,\alpha_2]}[0,1-x_1] \cdots R^{[\mu,\alpha_N]}[0,1-\sum_{i=1}^{N-1} x_i] f(x_1, \ldots, x_N) \quad (3.32)
\]

**Theorem 3.19.** Let \( f : T_N \to \mathbb{R} \) be \( D + 1 \) times continuously differentiable on \( T_N \), then the error functional representation of the \( \mu \)-panel offset trapezoidal rule operator for simplex \( R^{[\mu,\alpha]}T_N \) is of the form

\[
R^{[\mu,\alpha]}T_N f - IT_N f = \sum_{q=1}^{D} \frac{A_q}{\mu^q} + E_{D+1},
\]

where \( E_{D+1} = O \left( \frac{1}{\mu^{D+1}} \right) \)

**Proof.** See Doncker [1979].

More importantly, it can be shown [Doncker 1979] that \( R^{[\mu,\alpha]}T_N \) has an expansion in even powers of \( \mu \) when \( \mu \) or 2\( \mu \) is integer and \( t_\alpha \) is 0 or \( \frac{1}{2} \) with all \( i = 1, \ldots, N \). Then (3.31) applies and \( T^k_p \) has polynomial degree

\[
\begin{cases} 
2p + 2 - N, & \text{if } \{ \mu_k \}_k \in \mathbb{N} \\
2p + 1 - N, & \text{if } \{ 2\mu_k \}_k \in \mathbb{N}
\end{cases}
\]

Using \( \mu \)-panel offset trapezoidal rules and Romberg extrapolation, we generated one, three, five and seven degree \( \mu \)-panel offset trapezoidal rules for two, three- and four-dimensional simplices. Derivations of the rules can be found in Appendix A.1. These rules coincide with \( N \)-dimensional Grundmann-M"oller integration rules for the simplex, which are given in Tables 3.1 - 3.4. Grundmann and M"oller derived a family of cubature rules that are affine invariant for the simplex \( T_N \) using combinatorial methods [Grundmann and M"oller 1978]. The rules are embedded in dimension \( N \),
meaning that a higher degree rule contains the nodes of a lower degree rule. This property make them attractive because the error estimate is available naturally if you compare results of two formulas of different degree.

Let $s \in \mathbb{N}$. Then for odd degree $d = 2s + 1$, the Grundmann-Möller rule over $T_N$ is

$$G_s f := \sum_{i=0}^{s} 2^{-2s} (-1)^i \frac{(2s + 1 + N - 2i)^{2s+1}}{(2s + 1 + N - i)!i!} \times$$

$$\times \sum_{\sum_{j=0}^{N} \beta_j = i} \sum_{\beta_0 \geq \ldots \geq \beta_N} f \left( \frac{2\beta_0 + 1}{2s + 1 + N - 2i}, \ldots, \frac{2\beta_N + 1}{2s + 1 + N - 2i} \right) S,$$

(3.33)

where $\beta_i$ is a nonnegative integer for every $i$. The notation $\sum f((y))_S$ is a sum over all distinct permutations of $(N+1)$-vector $y = (1 - \sum_{i=1}^{N} y_i, y_1, \ldots, y_N)^T$. Rule (3.33) uses $\binom{N + s + 1}{s}$ function evaluations, which is relatively close to the lower bound of (3.9). One of the drawbacks is the many negative weights in the formula so that numerical stability may become an issue.

### 3.3 Adaptive methods

A problem with cubature methods using a set of fixed abscissas is that function evaluations might give us unimportant information about the integrand in regions where it behaves well and not enough information about it in important regions where variation is largest. Integrand with a dominant peak is an example of this kind of situation. To alleviate this problem there has been considerable amount of interest in adaptive integration methods. These methods determine the places of abscissas dynamically depending in some way on the integrand.

#### 3.3.1 Adaptive algorithms

One branch of adaptive integration methods is subregion-adaptive methods. In subregion-adaptive integration the integration region is divided into cells that have disjoint interiors and the integral and the error estimate are evaluated in each cell separately. The actual integration can be done using low-order polynomial formulas. If the total error estimate obtained by calculating the sum of all error estimates is not as low as desired, we try to reduce the error by dividing regions into subregions and repeating the procedure. Similar to extrapolation methods, adaptive algorithms can be used to construct powerful formulas, although not in the sense of polynomial
### Table 3.1: Degree 1 Grundmann-Möller cubature rule.

<table>
<thead>
<tr>
<th>Generator</th>
<th>Weight</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{1}{N+1}, \frac{1}{N+1}, \ldots, \frac{1}{N+1})</td>
<td>(\frac{1}{N!})</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table 3.2: Degree 3 Grundmann-Möller cubature rule.

<table>
<thead>
<tr>
<th>Generator</th>
<th>Weight</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{1}{N+1}, \frac{1}{N+1}, \ldots, \frac{1}{N+1})</td>
<td>(-\frac{(N+1)^3}{2^2(N+2)!})</td>
<td>1</td>
</tr>
<tr>
<td>(\frac{3}{N+3}, \frac{1}{N+3}, \ldots, \frac{1}{N+3})</td>
<td>(-\frac{(N+3)^3}{2^2(N+3)!})</td>
<td>N+1</td>
</tr>
</tbody>
</table>

### Table 3.3: Degree 5 Grundmann-Möller cubature rule.

<table>
<thead>
<tr>
<th>Generator</th>
<th>Weight</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{1}{N+1}, \frac{1}{N+1}, \ldots, \frac{1}{N+1})</td>
<td>(-\frac{(N+1)^5}{2^4(N+3)!2!})</td>
<td>1</td>
</tr>
<tr>
<td>(\frac{3}{N+3}, \frac{1}{N+3}, \ldots, \frac{1}{N+3})</td>
<td>(-\frac{(N+3)^5}{2^4(N+4)!})</td>
<td>N+1</td>
</tr>
<tr>
<td>(\frac{5}{N+5}, \frac{1}{N+5}, \ldots, \frac{1}{N+5})</td>
<td>(-\frac{(N+5)^5}{2^4(N+5)!})</td>
<td>N+1</td>
</tr>
<tr>
<td>(\frac{3}{N+5}, \frac{3}{N+5}, \frac{1}{n+5}, \ldots, \frac{1}{N+5})</td>
<td>(-\frac{(N+5)^5}{2^4(N+6)!})</td>
<td>(\frac{(N+1)!}{(N-1)!2!})</td>
</tr>
</tbody>
</table>

### Table 3.4: Degree 7 Grundmann-Möller cubature rule.

<table>
<thead>
<tr>
<th>Generator</th>
<th>Weight</th>
<th>Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{1}{N+1}, \frac{1}{N+1}, \ldots, \frac{1}{N+1})</td>
<td>(-\frac{(N+1)^7}{2^6(N+4)!3!})</td>
<td>1</td>
</tr>
<tr>
<td>(\frac{3}{N+3}, \frac{1}{N+3}, \ldots, \frac{1}{N+3})</td>
<td>(-\frac{(N+3)^7}{2^6(N+5)!2!})</td>
<td>N+1</td>
</tr>
<tr>
<td>(\frac{5}{N+5}, \frac{1}{N+5}, \ldots, \frac{1}{N+5})</td>
<td>(-\frac{(N+5)^7}{2^6(N+6)!})</td>
<td>N+1</td>
</tr>
<tr>
<td>(\frac{3}{N+5}, \frac{3}{N+5}, \frac{1}{N+5}, \ldots, \frac{1}{n+5})</td>
<td>(-\frac{(N+5)^7}{2^6(N+6)!})</td>
<td>(\frac{(N+1)!}{(N-1)!2!})</td>
</tr>
<tr>
<td>(\frac{7}{N+7}, \frac{1}{N+7}, \ldots, \frac{1}{N+7})</td>
<td>(-\frac{(N+7)^7}{2^6(N+7)!})</td>
<td>N+1</td>
</tr>
<tr>
<td>(\frac{5}{N+7}, \frac{3}{N+7}, \ldots, \frac{1}{N+7})</td>
<td>(-\frac{(N+7)^7}{2^6(N+7)!})</td>
<td>(\frac{(N+1)!}{(N-1)!})</td>
</tr>
<tr>
<td>(\frac{3}{N+7}, \frac{3}{N+7}, \frac{3}{n+7}, \frac{1}{N+7}, \ldots, \frac{1}{N+7})</td>
<td>(-\frac{(N+7)^7}{2^6(N+7)!})</td>
<td>(\frac{(N+1)!}{(N-2)!3!})</td>
</tr>
</tbody>
</table>
CHAPTER 3. NUMERICAL INTEGRATION

accuracy, using low order formulas as building blocks. There are different strategies for deciding which cells to refine. Two of such are local and global strategies [Krommer and Ueberhuber 1998].

In local subdivision strategy we select eligible cells for refinement by weighting the error in all cells with the fraction that the individual cell $B_s$ is of the whole region $B$

$$\epsilon(B_s) := \frac{\text{vol}(B_s)}{\text{vol}(B)} \epsilon,$$  \hfill (3.34)

where $\epsilon$ is the requested error.

After we have calculated $\epsilon(B_s)$ for each region, we keep the regions $B_s$ eligible for refinement until region satisfies

$$E(f; B_s) \leq \epsilon(B_s).$$  \hfill (3.35)

Then the requested error bound is obtained if Eq. (3.35) holds for all regions $B_s \subset B$

$$\left| \sum_{B_s} Q(f; B_s) - I_w(f; B) \right| \leq \sum_{B_s} \left| (Q(f; B_s) - I_w(f; B_s)) \right|$$

$$\leq \sum_{B_s} E(f; B_s)$$

$$\leq \sum_{B_s} \epsilon(B_s)$$

$$\leq \sum_{B_s} \frac{\text{vol}(B_s)}{\text{vol}(B)} \epsilon$$

$$= \epsilon.$$

The global subdivision strategy differs from local in that all regions $B_s$ are always eligible for subdivision. The region with the largest, possibly weighted error estimate is subdivided. A few of the weights $a_i$ on a region $B_i$ that have been used in global subdivision strategy can be found in Shapiro [1984]. We compare the following weights in our tests in Chapter 4.

$$a_i^{(1)} = 1$$

$$a_i^{(2)} = \frac{\text{vol}(B_i)}{\text{vol}(B)}$$

$$a_i^{(3)} = \frac{Q(f; B_i)}{E(f; B_i)}$$

Readers interested in adaptive integration algorithms are referred to articles [Genz and Cools 2003], [Kahaner and Wells 1979] and [Genz and Kass 1997]. We have provided a pseudocode for subregion-adaptive integration in Algorithm 1 that is being used in our tests in Chapter 4. Our algorithm is not a traditional algorithm because our only stopping criteria for the algorithm is if maximum allowed amount of abscissas is used. Usually algorithms check for convergence based on the error estimate. We do not adopt this traditional way because we are interested in how well the algorithm performs on average with a fixed amount of abscissas.
Algorithm 1: Subregion-adaptive integration algorithm

Divide the region $B$ into $B_n$

for $j \leftarrow 1$ to $n$ do
    Evaluate $Q_j(f; B_j)$ and $E_j(f; B_j)$
    Add $(B_j, Q_j(f; B_j), E_j(f; B_j))$ into datastructure
end

$m := \text{number of individual abscissas}$

while $m \leq \text{maximum allowed abscissas}$ do
    $k = \arg\max_i a_i E_i(f; B_i)$
    Divide the region $B_k$ into $B_{kn}$
    Remove $(B_k, Q_k(f; B_k), E_k(f; B_k))$ from the datastructure
    for $j \leftarrow 1$ to $n$ do
        Evaluate $Q_{kj}(f; B_{kj})$ and $E_{kj}(f; B_{kj})$
        Add $(B_{kj}, Q_{kj}(f; B_{kj}), E_{kj}(f; B_{kj}))$ into datastructure
    end
    $m = m + \text{new individual abscissas}$
end

$Q(f; B) := \sum_i Q_i(f; B_j)$
$E(f; B) := \sum_i E_i(f; B_j)$

Adaptive integration in Bayesian estimation

As discussed in Section 2.4, in Bayesian estimation we often approximate integrals of the form

$$E[x|y_0] = \frac{\int x l(x; y_0) p(x) dx}{\int l(x; y_0) p(x) dx},$$

(3.36)

where $x \in \mathbb{R}^N$. This means that we actually have $N + 1$ integrals to solve (1 for the normalizing constant, $N$ for elements of $x$). When computing integrals of above form we are interested in the ratio of the integrals and we want to obtain information of the integrands at the same abscissas. We construct our compound integration rule by adapting it according to the reciprocal of the normalizing constant, because usually $l(x; y_0) p(x)$ dominates the behaviour of $x l(x; y_0) p(x)$, and use the same formula to compute all the integrals in the process. This is of course computationally desirable situation because the adaptation needs to be done only once for the integral and the same function evaluations can be used effectively. In Figure 3.3 we illustrate how an adaptive algorithm refines the rule according to the behaviour of an integrand that is a typical one occurring in the positioning problem.

In adaptive integration the amount of data of the integral grows quickly with the refinement rounds and because of this it is a desirable property that as much infor-
Figure 3.3: The development of linear approximation of $p(x|y)$
mation as possible can be used from one refinement round to another. We choose to use Freudenthal’s simplicial refinement algorithm for subregion adapting process. The algorithm is discussed in Section 3.3.2. It subdivides a simplex by halving all its edges, and natural integration formulas to use are the ones that use function values in midpoints of edges and vertices. In this case the abscissas introduced into formula stay in formulas with more points when the grid is further refined, only their weights change.

3.3.2 Simplicial grid refinement

In subregion adaptive integration, we need to adjust the grid by subdividing some of its elements into smaller ones, preferably of the same kind. An important part of these kinds of adaptive algorithms is the scheme used to subdivide an element. In this work the basic integration region is taken to be the simplex. Accordingly, we present Freudenthal’s $N$-dimensional simplex refinement algorithm following the discussion in Bey [2000]. Freudenthal’s algorithm is used to divide a simplex into $2^N$ subsimplices of the same volume.

The cost of more powerful approximating properties when using simplices instead of cubes is that the refinement procedure is more complicated and there is not as much existing theory available, although finite element methods used to solve partial differential equations are known to use simplicial meshes. When using theory that has been developed for finite element methods we can ignore some of the quality constraints induced on the mesh. This includes so called conforming meshes what means that every subsimplex of a simplex must belong to a neighbouring simplex or to the boundary of the mesh and so called hanging nodes are not allowed. A hanging node is a vertex of a simplex that belongs also to another simplex and is not a vertex of that simplex. Not demanding conforming meshes is not a problem for ordinary integration problems, although it could be a desirable property when integrating continuous functions. In this work we do not use conforming meshes.

Definition 3.20. A permutation $\pi$ of a set $A$ is a bijection $A \to A$. Let $A = \{1, \ldots, N\}$ be a finite set. Then the group, under functional composition $\circ$, $S_N$ of all permutations of $A$ is called the symmetric group on $A$.

We can observe that $S_N$ has $N!$ elements. For permutation $\pi$, its inverse function $\pi^{-1}$ is the permutation that reverses the direction of mapping $\pi$ that is, for $a \in A$, $a' = \pi^{-1}(a)$ such that $a = \pi(a')$.

Let $x_0, \ldots, x_N$ be the $N+1$ vertices of $N$-dimensional simplex. Then an affine invariant way to subdivide a simplex into subsimplices $T_{x_{0:k}, \pi}$ is Algorithm 2. The subdivision process is illustrated in Figure 3.4 for a tetrahedron.
Algorithm 2: Freudenthal’s algorithm [Bey 2000]

\[
\begin{align*}
\text{for } & \ 0 \leq k \leq N \text{ do} \\
& x_{0,k} := \frac{1}{2}(x_0 + x_k) \\
& \text{for } \pi \in S_N \text{ do} \\
& \quad \text{if } k = 0 \text{ or } k = N \text{ then} \\
& \quad \quad \text{if } \pi^{-1}(1) \leq \cdots \leq \pi^{-1}(N) \text{ then} \\
& \quad \quad \quad \text{for } 1 \leq l \leq N \text{ do} \\
& \quad \quad \quad \quad x_{l,k} := x_{l-1,k} + \frac{1}{2}(x_{\pi(l)} - x_{\pi(l)-1}) \\
& \quad \quad \quad \text{end} \\
& \quad \quad T_{x_{0,k},\pi} := [x_{0,k}, \ldots, x_{N,k}] \\
& \quad \text{end} \\
& \quad \text{else} \\
& \quad \quad \text{if } \pi^{-1}(1) \leq \cdots \leq \pi^{-1}(k) \text{ and } \pi^{-1}(k+1) \leq \cdots \leq \pi^{-1}(N) \text{ then} \\
& \quad \quad \quad \text{for } 1 \leq l \leq N \text{ do} \\
& \quad \quad \quad \quad x_{l,k} := x_{l-1,k} + \frac{1}{2}(x_{\pi(l)} - x_{\pi(l)-1}) \\
& \quad \quad \quad \text{end} \\
& \quad \quad T_{x_{0,k},\pi} := [x_{0,k}, \ldots, x_{N,k}] \\
& \quad \text{end} \\
& \text{end} \\
& \text{end}
\end{align*}
\]

Figure 3.4: Tetrahedron divided by Freudenthal’s algorithm
Because Freudenthal's algorithm results in $2^N$ subsimplices, the growth of number of simplices is relatively fast. Other subdivision methods exist and also ones that might be more feasible in some cases. The power of Freudenthal’s algorithm lies in the simplicity of the process and also the fact that it is a stable algorithm in the sense that recursive subdivisions using it result in simplices whose interior angles are bounded away from zero.

### 3.4 Monte Carlo methods

Sampling based methods such as Monte Carlo integration are commonly used to evaluate integrals and they are especially useful when dealing with high dimensional integrals.

The approximation methods considered so far have all been deterministic in the sense that the places of the abscissas have been chosen according to some predefined deterministic strategy. Another way of evaluating the integral is to use randomly selected abscissas. Monte Carlo provides a simple and direct tool for executing this. Used extensively in simulation, Monte Carlo is also used to evaluate integrals and in some cases it might actually be the only viable method of integration.

In Monte Carlo integration, we introduce some probability density function $p(x)$ that satisfies $\text{supp}(p) \subseteq \text{supp}(f)$ and reformulate the integral as

\[
If = \int_B f(x)dx = \int_{\mathbb{R}^N} \frac{f(x)}{p(x)}p(x)dx. \tag{3.37}
\]

If the introduced function $p$ is uniform density, then this method is sometimes referred to as the basic Monte Carlo integration. In the general case, $p$ is called an importance sampler and the whole procedure importance sampling. From Eq. (3.37) we see that we can interpret $If$ as the mean of $\frac{f(x)}{p(x)}$ with a probability density function $p$,$$
If = E \left[ \frac{f(x)}{p(x)} \right].$$

We get an estimate for the integral from estimating $E \left[ \frac{f(x)}{p(x)} \right]$ as a sample mean

$$
E \left[ \frac{f(x)}{p(x)} \right] \approx \frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)} = \sum_{i=1}^{n} \frac{1}{np(x_i)} f(x_i), \tag{3.38}
$$

where $x_1, \ldots, x_n$ are independent samples from the probability density $p$. The name importance sampling comes from choosing $p$ such that it would generate more samples in regions where the value of the function $f$ is important for the value of the integral.
[Evans and Swartz 2005]. Observing Eq. (3.38), we can see that it is similar to interpolatory formulas with weighted sum of function values. The only difference is that abscissas $x_i$ and weights $c_i = \frac{1}{np(x_i)}$ are realized values of random variables and $Q_{n,f}$ itself is a realization of a random variable

$$
\hat{Q}_{p,n,f} := \frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)}.
$$

Almost sure convergence of $\hat{Q}_{p,n,f}$ to $If$ is guaranteed by the strong law of large numbers

$$
P(\lim_{n \to \infty} \hat{Q}_{p,n,f} = If) = 1.
$$

It can be shown that $\hat{Q}_{p,n,f}$ is an unbiased estimate of $If$. This means that for any $n$

$$
E[\hat{Q}_{p,n,f}] = E\left[\frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)}\right] = \frac{1}{n} \sum_{i=1}^{n} E\left[\frac{f(x_i)}{p(x_i)}\right] = E\left[\frac{f}{p}\right].
$$

To be of use as an integration method, we need to express an error estimate for Eq. (3.39) that is,

$$
\hat{E}_{p,n}(f; \mathbb{R}^N) := |\hat{Q}_{p,n,f} - If|.
$$

Because the function is sampled in random points, a deterministic error estimate is not available. To estimate the error, the probability density of $\hat{E}_{n}(f; \mathbb{R}^N)$ has to be examined. Let the variance $\sigma^2$ of $f/p$ be finite. Then

$$
\sigma^2(\hat{Q}_{p,n,f}) = \frac{\sigma^2}{n} < \infty,
$$

where

$$
\sigma^2 = \int_{\mathbb{R}^N} \left( \frac{f(x)}{p(x)} - If \right)^2 p(x)dx.
$$

By the central limit theorem we obtain

$$
\lim_{n \to \infty} \sqrt{n}(\hat{Q}_{p,n,f} - If) \sim N(0, \sigma^2),
$$

where $N(0, \sigma^2)$ is Gaussian distribution. This gives the importance sampling algorithm the convergence rate of $O(\frac{1}{\sqrt{n}})$. Although the convergence rate is rather slow, one of the strengths of Monte Carlo integration lies in that the convergence rate does not depend on the dimension of the integral.

Because we are unable to integrate $If$, we need an estimator for $\sigma^2$. A natural one is the sample mean

$$
\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{f(x_i)}{p(x_i)} - \hat{Q}_{p,n,f} \right)^2 = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{f(x_i)}{p(x_i)} \right)^2 - \frac{1}{n^2} \left( \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)} \right)^2.
$$
Combining (3.43), (3.45) and (3.46) we obtain the error estimate for Monte Carlo integration

\[
\hat{\sigma} (\hat{Q}_{p,n} f) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left( \frac{f(x_i)}{p(x_i)} \right)^2 - \frac{1}{n^2} \left( \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)} \right)^2}.
\]  

(3.47)

As mentioned before, the name importance sampling refers to choosing \( p \) so that the points generated from it would be more dense in important areas for the integral. This makes the choice of \( p \) the most important aspect of importance sampling [Evans and Swartz 2005]. If the function \( f \) is positive on \( \Omega \), then it is the choice \( p(x) := \frac{f(x)}{p(x)} \) that would give the optimal variance \( \sigma^2 (\hat{Q}_p) = 0 \). It is important to notice that to choose this \( p \), the value of the solution \( I f \) would have to be known. Usually the best choices for \( p \) are the ones that mimic the behaviour of \( f \) so that \( \frac{f(x)}{p(x)} \) is close to a constant function.

As observed in Section 2.4, the integrals in Bayesian estimation problems are usually of the form (2.18). This means that really we are interested in the ratio of the integrals

\[
\mu := E[x|y] = \frac{\int x l(x; y)p(x)dx}{\int l(x; y)p(x)dx}.
\]  

(3.48)

To compute this ratio we again introduce an importance sampler, which we denote now with \( p_i \) and estimate \( \mu \) by

\[
\hat{\mu}_n = \frac{\sum_{i=1}^{n} \frac{x_i l(x_i; y)p(x_i)}{p_i(x_i)}}{\sum_{i=1}^{n} \frac{l(x_i; y)p(x_i)}{p_i(x_i)}},
\]  

(3.49)

where \( x_i \)'s are the same in the numerator and denominator. The importance sampler should be chosen according to the behaviour of \( l(x; y)p(x) \) when it dominates the behaviour of \( x l(x; y)p(x) \).

**Theorem 3.21.** Let \( g(x) = x l(x; y)p(x) \) and \( f(x) := l(x; y)p(x) \). Let \( p_i(x) \) be an importance sampler.

If \( I f \neq 0 \) and \( \text{Var} \left( \frac{f(x)}{p_i(x)} \right), \text{Var} \left( \frac{g(x)}{p_i(x)} \right) < \infty \), then

\[
\lim_{n \to \infty} \sqrt{n} (\hat{\mu}_n - \mu_n) \sim N(0, \sigma^2)
\]

where

\[
\sigma^2 = \frac{1}{(I f)^2} \left\{ \text{Var} \left[ \frac{g(x)}{p_i(x)} \right] + \frac{(I g)^2}{(I f)^2} \text{Var} \left[ \frac{f(x)}{p_i(x)} \right] - 2 \frac{I g}{I f} \text{Cov} \left[ \frac{g(x)}{p_i(x)}, \frac{f(x)}{p_i(x)} \right] \right\}
\]  

(3.50)


The estimator \( \hat{\sigma}^2 \) of \( \sigma^2 \) is obtained via using sampled analogs of the integrals in Theorem 3.21 [Evans and Swartz 2005].
Random numbers

To use importance sampling we have to generate samples from $p$, which can be a daunting task. When implementing Monte Carlo methods truly random numbers are not possible to generate but instead algorithmically generated numbers have to be used. These sequences consist of so called pseudo-random numbers. Pseudo-random sequences are designed to simulate samples from truly random sequences from different probability densities. Readers interested in pseudo-random number generating algorithms are referred to Niederreiter [1992].

Often random number generators are used to generate uniformly distributed numbers on the positive unit cube. But as mentioned in previous sections, the simplex is more versatile integration region than cube. To use basic Monte Carlo integration over simplex, we need to generate random numbers from uniform distribution on a simplex. The following theorem reveals the formula for generating uniformly distributed random numbers on an $N$-dimensional simplex.

**Theorem 3.22.** [Devroye 1986]. Let $(S_0, \ldots, S_N)$ be the spacings between $N$ uniform samples on $[0, 1]$. Then

$$x = \sum_{i=0}^{N} S_i x_i$$

is uniformly distributed in the simplex $T_N$ defined by vertices $(x_0, \ldots, x_N)$.

**Proof.** See Devroye [1986] page 568.

In a polygonal region, which can be divided into a collection of simplices, we generate an uniform sample by first choosing a simplex from the collection with a probability proportional to the volume of the simplex to the volume of the whole region. After this we generate an uniform sample inside the chosen simplex. [Devroye 1986].
Chapter 4

Testing and Simulations

In this chapter we provide numerical results that show how different integration methods perform when applied to the positioning problem. First in Section 4.1 we introduce the basic integration algorithms used. Because we are unable to provide exact positions to which we could compare different performances, we choose to solve a reference position with a computationally intensive integration algorithm. This is discussed in Section 4.1.1. In the Section 4.2 we provide the results of the tests.

### 4.1 Compared integration methods

In the testing part we compared the performance of adaptive algorithm using different basic integration rules and the basic Monte Carlo integration algorithm. For the adaptive algorithm, which used a simplex as a basic cell, our basic integration rules are listed in Table 4.1 in forms which are for simplex $T_N$. Formulas in the table are given in the form

$$Q_n = \sum c_i \sum f((x_i))_S,$$

where $x_i$ are the given generators and $c_i$ corresponding weights and the sum $\sum f((y))_S$ is over all distinct permutations of $N + 1$-vector $y = (1 - \sum_{i=1}^{N} y_i, y_1, \ldots, y_N)^T$. 
The formulas $Q_{n_2}$ for two-, three- and four-dimensional simplices are derived in Section 3.2, Section 3.2.1 and in Appendix A.1.

The function $\xi(x)$ is the number of different simplices into which $x$ belongs after the simplex has been subdivided by Freudenthal’s algorithm.

The error estimates were taken to be

$$E_{n_1}(f; B) = |Q_{n_1}(f; B) - Q_{n_0}(f; B)|$$  \hspace{1cm} (4.2)

$$E_{n_2}(f; B) = |Q_{n_2}(f; B) - Q_{n_0}(f; B)|$$  \hspace{1cm} (4.3)

$$E_{n_7}(f; B) = |Q_{n_7}(f; B) - Q_{n_5}(f; B)|,$$  \hspace{1cm} (4.4)

where

$$Q_{n_0}(f; T_N) = \frac{1}{N + 1} \sum_{k=1}^{N} f((1, 0, \ldots, 0))_S$$  \hspace{1cm} (4.5)

is a degree one rule using the vertices of simplex as abscissas and $Q_{n_5}$ is the 5th degree Grundmann-Möller rule as given in Table 3.3.

| 2D | Symbol | generator | weight | generator | weight | d 
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{n_1}$</td>
<td>$(1, 0, 0)$</td>
<td>$\frac{1}{24}$</td>
<td>$(\frac{1}{2}, \frac{1}{2}, 0)$</td>
<td>$\frac{1}{8}$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$Q_{n_2}$</td>
<td>$(1, 0, 0)$</td>
<td>0</td>
<td>$(\frac{1}{2}, \frac{1}{2}, 0)$</td>
<td>$\frac{1}{6}$</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$Q_{n_7}$</td>
<td>See Table 3.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3D</th>
<th>Symbol</th>
<th>generator</th>
<th>weight</th>
<th>generator</th>
<th>weight</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{n_1}$</td>
<td>$(1, 0, 0, 0)$</td>
<td>$\frac{1}{32}$</td>
<td>$(\frac{1}{2}, \frac{1}{2}, 0, 0)$</td>
<td>$\frac{1}{32} \xi(x)$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$Q_{n_2}$</td>
<td>$(1, 0, 0, 0)$</td>
<td>$\frac{1}{120}$</td>
<td>$(\frac{1}{2}, \frac{1}{2}, 0, 0)$</td>
<td>$\frac{1}{30}$</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$Q_{n_7}$</td>
<td>See Table 3.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4D</th>
<th>Symbol</th>
<th>generator</th>
<th>weight</th>
<th>generator</th>
<th>weight</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{n_1}$</td>
<td>$(1, 0, 0, 0, 0)$</td>
<td>$\frac{1}{120}$</td>
<td>$(\frac{1}{2}, \frac{1}{2}, 0, 0, 0)$</td>
<td>$\frac{1}{120} \xi(x)$</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$Q_{n_2}$</td>
<td>$(1, 0, 0, 0, 0)$</td>
<td>$\frac{1}{360}$</td>
<td>$(\frac{1}{2}, \frac{1}{2}, 0, 0, 0)$</td>
<td>$\frac{1}{180}$</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$Q_{n_7}$</td>
<td>See Table 3.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: Compared interpolatory integration methods.

We implemented a global subregion adaptive integration algorithm in Matlab and compared different weights for different subdivision strategies. The weights considered were

$$a_i^{(1)} = 1$$

$$a_i^{(2)} = \frac{\text{vol}(B_i)}{\text{vol}(B)}$$  \hspace{1cm} (4.6)

$$a_i^{(3)} = \frac{|Q(f; B_s)|}{E(f; B_s)}.$$

We tested the performance of the algorithm in dimensions 2 – 4 with a set of testfunctions. Testfunctions were summed Gaussian peaks on the positive hypercube.

$$f(x) = \sum_{j=1}^{2} \exp \left( -\sum_{i=1}^{N} \eta_{ij}^2 (x_i - \zeta_{ij})^2 \right)$$  \hspace{1cm} (4.7)
The test functions have parameters $\eta, \zeta \in \mathbb{R}^{N \times k}$ that determine the difficulty of the integration problem and the locations of the $k$ peaks respectively. We can express the integral of $f$ over $C_N^+$ using error functions in the form

$$
\int_{C_N^+} f(x)dx = \sum_{j=1}^{2} \prod_{i=1}^{N} \sqrt{\frac{\pi}{2\eta_{ij}}} \left( \text{erf}(\eta_{ij}(1 - \zeta_{ij})) + \text{erf}(\eta_{ij}\zeta_{ij}) \right),
$$

(4.8)

which we can evaluate very accurately.

We generated 100 different test cases where the locations of the peaks were randomly generated inside $N$-dimensional positive unit cube and the difficulty parameter were randomly generated on the interval $6 - 8$.

The convergence results are illustrated in Figures 4.1 - 4.3. In the figure we have plotted the mean of the relative error of the hundred test cases against the number of abscissas. The weight $a_i^{(2)}$ based on the relative size of the region seemed to systematically give the best results on average, so we chose it as our weight $a_i$ on all different adaptive methods. For $Q_{n7}$ we have plotted the results up to very large number of abscissas, because it is used to compute the reference solutions of the position tests.

![Figure 4.1: Convergence of adaptive algorithm using different adaption strategies and basic rule $Q_{n1}$.](image)

4.1.1 Reference solutions

The reference solutions for our positioning tests were computed numerically using the subregion adaptive integration algorithm. A very large number of function evaluations were allowed to obtain accurate references. We used the rule pair $Q_{n7}, Q_{n5}$ as our basic integration rule. The results are presented in Table 4.2. From the results it could be seen that in two-dimensional case we would obtain extremely accurate reference solutions by dividing the region adaptively into approximately 1500 simplices, which amounts to $19 \cdot 1500 = 28500$ abscissas when using Grundmann-Möller 7-5 pair. In
three and four dimensions, we can see the dimensionality effect as the division of the region into 5000 and 15000 simplices (175000 and 840000 abscissas respectively) result in reported accuracies in Table 4.2.

### 4.2 Positioning tests

To compare the performance of different integration methods applied to positioning problem using for example cellular networks, we generated 400 test integrals from 15 different scenarios in dimensions 2 – 4. Integration region $B$ was taken to be bounded with sector information in a plane. Figure 4.4 illustrates different geometries.

In three- and four-dimensional tests we added perpendicular constraints $\|x_3\|_\infty < 100$ and $\|x_{3:4}\|_\infty < 100$ to the region. A random target was generated in the location...
<table>
<thead>
<tr>
<th></th>
<th>2D</th>
<th></th>
<th>3D</th>
<th></th>
<th>4D</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Relative error</td>
<td>Times (%)</td>
<td>Relative error</td>
<td>Times (%)</td>
<td>Relative error</td>
<td>Times (%)</td>
<td></td>
</tr>
<tr>
<td>$&lt; 10^{-12}$</td>
<td>10</td>
<td>$&lt; 10^{-8}$</td>
<td>12</td>
<td>$&lt; 10^{-7}$</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>$&lt; 10^{-11}$</td>
<td>49</td>
<td>$&lt; 10^{-7}$</td>
<td>52</td>
<td>$&lt; 10^{-6}$</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>$&lt; 10^{-10}$</td>
<td>97</td>
<td>$&lt; 10^{-6}$</td>
<td>98</td>
<td>$&lt; 10^{-5}$</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>$&lt; 10^{-9}$</td>
<td>100</td>
<td>$&lt; 10^{-5}$</td>
<td>100</td>
<td>$&lt; 10^{-4}$</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: The results of $Q_{n_T}(f, C_N^\pm)$

Figure 4.4: Sector information bounded integration regions

(a) Region 1  
(b) Region 2  
(c) Region 3
θ from uniform distribution inside the cell and Gaussian random errors were added to the range measurements between base stations and the target. The noise in the range measurements was taken to have a standard deviation generated from [50, 100] or [100, 150], depending on the scenario. In three dimensional tests, we added an altitude measurement that had Gaussian error with mean 0 and standard deviation generated from [25, 35], [50, 60] or [75, 85]. In four dimensional tests, we added a velocity measurement, that had Gaussian error with mean 0 and standard deviation generated from [15, 25], [30, 40] or [45, 55]. The different test cases are collected in Table 4.3. The cases with smaller variances are considered to be more difficult integrands since they usually result in functions where most of the mass is concentrated in a relatively small region, which emphasizes the dimensionality effect. When measurement errors are normally distributed random variables, the measurement likelihood is

\[ l(x; r) = \exp \left( -\frac{1}{2} \sum_{i=1}^{k} \frac{\|s_i - x\| - r_i}{\sigma_i^2} \right), \] (4.9)

where \( s_i \) is the location of the \( i^{th} \) base station. In half of our test cases we take a uniform prior in \( B \) and in the other half we take a Gaussian prior with mean \( \theta \) and standard deviation between [200, 300], [50, 100] or [25, 75] in two, three and four dimensional cases respectively.

We took the expectation of the posterior to be the position estimate so that the integrals were

\[ E[x|r] = \frac{\int_B x l(x;r)p(x)dx}{\int_B l(x;r)p(x)dx} \] (4.10)

<table>
<thead>
<tr>
<th># Range measurements</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sigma_1 ) ∈</td>
<td>50, 100</td>
<td>100, 150</td>
<td>50, 100</td>
<td>50, 100</td>
<td>100, 150</td>
</tr>
<tr>
<td>( \sigma_2 ) ∈</td>
<td>-</td>
<td>-</td>
<td>50, 100</td>
<td>100, 150</td>
<td>100, 150</td>
</tr>
<tr>
<td>3D: ( \sigma_3 ) ∈</td>
<td>25, 35</td>
<td>75, 85</td>
<td>25, 35</td>
<td>50, 60</td>
<td>75, 85</td>
</tr>
<tr>
<td>4D: ( \sigma_3, \sigma_4 ) ∈</td>
<td>15, 25</td>
<td>45, 55</td>
<td>15, 25</td>
<td>30, 40</td>
<td>45, 55</td>
</tr>
</tbody>
</table>

Table 4.3: Test cases

The reference expectation was computed with the adaptive cubature algorithm using large number of function evaluations. The algorithm used a basic cubature rule exact for 7th degree polynomials that in Section 4.1.1 was shown to often give reliable results. It is important to notice that we are comparing our methods against approximate solutions, so that a method that gives an answer closest to the reference position is not necessarily the most accurate method but a method that gives an estimate closest to an estimate that can be assumed to be very accurate.
4.2.1 2D-tests

In two-dimensional tests, we estimated the integrals using different number of abscissas. Table 4.4 gives the percentage of how many times a given method gave the estimate closest to the reference estimate out of all the methods. The tests show that, as expected, the basic Monte Carlo integration does not work as well as adaptive interpolatory methods with a low amount of points. When considering adaptive interpolatory methods, we see that the low order formulas $Q_{n_1}$ and $Q_{n_2}$ perform better than the high order formula $Q_{n_7}$ with a small number of points. This is because when the formulas use the same number of points, the low order formulas that use small number of abscissas per simplex have adapted themselves according to the behavior of the integrand more times than the higher degree formula that uses large number of points per simplex. When the large of abscissas of the rule goes up, the higher degree formula has adapted itself to the integrand and from this point on it gives the best results on average. The trend is clear in Table 4.4. It shows that interpolatory methods converge faster than Monte Carlo and higher degree formulas converge faster than lower degree formulas.

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>pts</td>
<td>$Q_{n_1}$ $Q_{n_2}$ $Q_{n_7}$ mc</td>
<td>$Q_{n_1}$ $Q_{n_2}$ $Q_{n_7}$ mc</td>
<td>$Q_{n_1}$ $Q_{n_2}$ $Q_{n_7}$ mc</td>
<td>$Q_{n_1}$ $Q_{n_2}$ $Q_{n_7}$ mc</td>
<td>$Q_{n_1}$ $Q_{n_2}$ $Q_{n_7}$ mc</td>
</tr>
<tr>
<td>200</td>
<td>28 59 12 1</td>
<td>7 61 32 0</td>
<td>32 58 7 3</td>
<td>29 62 7 2</td>
<td>20 60 20 0</td>
</tr>
<tr>
<td>400</td>
<td>11 76 13 0</td>
<td>2 50 48 0</td>
<td>18 64 18 0</td>
<td>15 69 16 0</td>
<td>7 56 38 0</td>
</tr>
<tr>
<td>600</td>
<td>5 64 31 0</td>
<td>1 32 67 0</td>
<td>10 56 33 0</td>
<td>8 62 30 0</td>
<td>3 40 58 0</td>
</tr>
<tr>
<td>800</td>
<td>2 58 41 0</td>
<td>0 27 73 0</td>
<td>6 50 44 0</td>
<td>4 52 44 0</td>
<td>1 25 74 0</td>
</tr>
<tr>
<td>1000</td>
<td>2 46 52 0</td>
<td>0 11 89 0</td>
<td>4 43 53 0</td>
<td>3 42 55 0</td>
<td>0 17 83 0</td>
</tr>
<tr>
<td>1200</td>
<td>1 44 55 0</td>
<td>0 9 91 0</td>
<td>3 35 62 0</td>
<td>1 36 63 0</td>
<td>0 15 85 0</td>
</tr>
<tr>
<td>1400</td>
<td>0 41 58 0</td>
<td>0 7 93 0</td>
<td>1 32 67 0</td>
<td>1 25 74 0</td>
<td>0 11 89 0</td>
</tr>
<tr>
<td>1600</td>
<td>0 36 64 0</td>
<td>0 5 95 0</td>
<td>1 29 70 0</td>
<td>1 23 77 0</td>
<td>0 7 93 0</td>
</tr>
<tr>
<td>1800</td>
<td>0 29 70 0</td>
<td>0 3 97 0</td>
<td>1 22 77 0</td>
<td>0 21 79 0</td>
<td>0 6 94 0</td>
</tr>
</tbody>
</table>

Table 4.4: 2D: How many times (%) each method gave the best answer

In Figure 4.5 we present some details on how the error of the estimate is divided into different intervals. The results show that formulas using 200 abscissas constructed using low order basic integration rules give us estimates that are almost always inside 5m circle centered at the reference estimate of the position.

4.2.2 3D-tests

For three-dimensional tests, we added the vertical position dimension to our tests. The tests were formed by adding an altitude measurement to the two-dimensional case. The altitude measurement was taken to have Gaussian error with mean 0 and randomly generated standard deviation $\sigma_3 \in [25, 35], [50, 60]$ or $[75, 85]$, depending on the test case. When considering the error of the estimate, we did it by comparing the projections of the positions on a two dimensional plane.
Compared to two-dimensional tests, Table 4.5 shows that with low amount of abscissas $Q_{n_1}$ give better results than $Q_{n_2}$ which is due to the negative weights in formula $Q_{n_2}$. Increasing the amount of simplices in the region, the local behaviour of the integrand is better approximated with second order polynomials so that the influence of negative weights decrease. The same is true when looking at the performance of $Q_{n_7}$. In three dimensions it clearly behaves much worse with a low number of abscissas than the low order formulas. This is due to the fact that in three dimensions $Q_{n_7}$ uses already 35 abscissas in a simplex and because when simplex is partitioned into 8 subsimplices, we introduce $7 \cdot 35 = 245$ extra abscissas into the formula in each iteration. Monte Carlo integration is performing a little better compared to two-dimensional tests but still interpolatory methods offer much better accuracies. The trend is clear from Table 4.5. It shows that interpolatory rules converge faster than Monte Carlo, and higher degree formulas converge faster than lower degree formulas.

In Figure 4.6 we provide detailed results of tests when using approximately 700 abscissas. Accuracies drop considerably when compared to the two dimensional tests.
CHAPTER 4. TESTING AND SIMULATIONS

Table 4.5: 3D: How many times (%) each method gave the best answer

<table>
<thead>
<tr>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>pts</td>
<td>$Q_{n_1}$</td>
<td>$Q_{n_2}$</td>
<td>$Q_{n_7}$</td>
<td>mc</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>23</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>700</td>
<td>53</td>
<td>35</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>1100</td>
<td>48</td>
<td>43</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>1500</td>
<td>42</td>
<td>50</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>1900</td>
<td>39</td>
<td>54</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>2300</td>
<td>31</td>
<td>60</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>2700</td>
<td>25</td>
<td>65</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>3100</td>
<td>23</td>
<td>68</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>3500</td>
<td>21</td>
<td>69</td>
<td>9</td>
<td>1</td>
</tr>
</tbody>
</table>

But still lower order formulas often give an estimate inside 20m radius from the reference position.

4.2.3 4D-tests

For four dimensional tests, we added two velocity dimensions to our two dimensional tests. An independent zero velocity measurement was added to (4.9). The errors in the new measurements were Gaussian with mean 0 and variance generated randomly from the intervals $[15, 25], [30, 45]$, and $[45, 55]$, depending on the test case. The error of the estimate was taken to be the error in the position dimensions.

The results show that the dimensionality effect favors Monte Carlo integration much more clearly in four dimensions. The detailed results in Figure 4.7 show that with 2000 points it is as reliable as interpolatory methods with much less computational effort, and in region 3 tests it performs best. Notice that $Q_{n_7}$ does not provide a result with 2000 abscissas in region 3 tests. This is because in four dimensions, the region is divided into so many simplices from the start that $Q_{n_7}$ which uses 56 abscissas per simplex, has over 2000 abscissas even when no refinement rounds is performed.

Table 4.6: 4D: How many times (%) each method gave the best answer

Table 4.6 shows how the results changed when more abscissas were introduced into the formulas. The behaviour is similar to three dimensional cases. Interpolatory methods
converge faster than Monte Carlo, and higher degree formulas converge faster that lower degree formulas. $Q_{n_1}$ is the strongest first because all its weights are positive.
Figure 4.7: 4D: 2000 abscissas
Chapter 5

Conclusions and Future Work

In this thesis we discussed the theory of numerical integration methods and Bayesian framework for local positioning through literacy review. In addition, we compared the performance of simplicial subregion adaptive integration methods to Monte Carlo integration in positioning scenarios. This was done to see whether there is a reason for using sophisticated methods to solve integrals in scenarios where the speed of the method and computational requirements are considered more important than usually. Different basic rules in adaptive algorithm were compared to see which one would give the best results on average.

The tests show that the adaptive algorithm favors low dimensional basic integration rules to higher ones due to more subregion refinement rounds we get when using the number of abscissas as a termination criteria. It was shown that for integrals occurring in positioning that have certain distinct features such as multiple peaks and curved ridges, the subdivision strategy of the adaptive algorithm is an important factor in the performance.

In two- and three-dimensional tests interpolatory methods provide much better results with low number of function evaluations than Monte Carlo and can be considered better for the problem. In four dimensional tests, the dimensionality effect favors Monte Carlo. This conclusions is based on that even interpolatory methods need a
high amount of function evaluations to obtain considerably better results than Monte Carlo which will always be computationally much more feasible method.

The promising results in two and three dimensional tests show that there is a reason for further studies of adaptive integration in this particular application. Future work include studying different subdivision methods for simplices and different basic integration rules, preferably rules that contain only positive weights. Also, the possibility of separating the position and velocity integrals, so that we would have at most three-dimensional integrals, should be studied.

Different Monte Carlo methods would also have to be researched. If even the basic Monte Carlo integration gives results competitive in four dimensional test case, it would be interesting to compare Monte Carlo methods using importance and stratified sampling, or even adaptive Monte Carlo methods.

For additional future work, optimization of the integration programs would have to be done and extensive comparison to see the actual computational requirements of the methods. Current Matlab implementations of the algorithms do not give a correct picture of the relative performances.
Bibliography


FCC. FCC adopts rules to implement enhanced 911 for wireless services. CC docket 94-102, FCC News, 1996.


Appendix A

Cubature Rules

A.1 Interpolatory Cubature Rules

According to the theory discussed in Section 3.2.2 we derive cubature formulas in dimensions $2-4$ using degree one $\mu$-panel offset trapezoidal formulas and extrapolate them using the recursion formula

\[
\begin{align*}
T_0^k &= R[\mu_k, \alpha][0,1] f \\
T_p^k &= T_{p-1}^{k+1} + \mu_{k,p} (T_{p-1}^{k+1} - T_{p-1}^k),
\end{align*}
\]

which gives algebraic degree $d = p + 2 - N$ formulas if $\{\mu_k\}_k$ is a sequence of integers and degree $d = 2p + 1 - N$ formulas if $\{2\mu_k\}_k$ is a sequence of integers. $\mu_{k,p} = \frac{\mu_k^2}{\mu_{k+p} - \mu_k}$ when $R[\mu_k, \alpha][0,1] f$ has an even expansion in powers $\mu^{-1}$.

In the formulas the notation $\sum f(x)_S$ means that the sum is over all distinct permutations of $x$. 

Two Dimensional Rules

\[ T_0^0 = R_{x_1}^{[1/2,0]}[0, 1] R_{x_2}^{[1/2,0]}[0, 1 - x_1] f(x_1, x_2) \]

\[ = R_{x_1}^{[1/2,0]}[0, 1] \left[ \frac{1}{1/2} \sum_{j=-\infty}^{\infty} [H(2 - 2j - x_1) - H(1 - 2j)] f(x_1, 2j - 1) \right] \]

\[ = R_{x_1}^{[1/2,0]}[0, 1] [2H(-x_1) f(x_1, 1)] \]

\[ = R_{x_1}^{[1/2,0]}[0, 1] g(x_1) \]

\[ = \frac{1}{1/2} \sum_{j=-\infty}^{\infty} [H(2 - 2j) - H(1 - 2j)] g(2j - 1) \]

\[ = 2 [H(0) g(1)] = 2 [H(0) H(-1) f(1, 1)] \]

\[ = 0 \]

\[ T_0^1 = R_{x_1}^{[3/2,0]}[0, 1] R_{x_2}^{[3/2,0]}[0, 1 - x_1] f(x_1, x_2) \]

\[ = R_{x_1}^{[3/2,0]}[0, 1] \left[ \frac{1}{3/2} \sum_{j=-\infty}^{\infty} \left[ H \left( 1 - x_1 - \frac{2j - 1}{2 \cdot 3/2} \right) - H \left( \frac{1 - 2j}{2 \cdot 3/2} \right) \right] f \left( x_1, \frac{2j - 1}{2 \cdot 3/2} \right) \right] \]

\[ = R_{x_1}^{[3/2,0]}[0, 1] \left[ \frac{2}{3} \sum_{j=-\infty}^{\infty} \left[ H \left( \frac{4 - 2j}{3} - x_1 \right) - H \left( \frac{1 - 2j}{3} \right) \right] f \left( x_1, \frac{2j - 1}{3} \right) \right] \]

\[ = R_{x_1}^{[3/2,0]}[0, 1] \left[ \frac{2}{3} \left[ H \left( \frac{2}{3} - x_1 \right) f \left( x_1, \frac{1}{3} \right) + H \left( \frac{0}{3} - x_1 \right) f(x_1, 1) \right] \right] \]

\[ = R_{x_1}^{[3/2,0]}[0, 1] g(x_1) \]

\[ = \frac{2}{3} \sum_{j=-\infty}^{\infty} \left[ H \left( \frac{4 - 2j}{3} \right) - H \left( \frac{1 - 2j}{3} \right) \right] g \left( \frac{2j - 1}{3} \right) \]

\[ = \frac{2}{3} \left[ H \left( \frac{2}{3} \right) g \left( \frac{1}{3} \right) + H(0) g(1) \right] \]

\[ = \frac{2}{3} \left[ \frac{2}{3} \left[ H \left( \frac{1}{3} \right) f \left( \frac{1}{3}, \frac{1}{3} \right) + H \left( -\frac{1}{3} \right) f \left( \frac{1}{3}, 1 \right) \right] \right] + \frac{1}{2} \frac{2}{3} \left[ H \left( -\frac{1}{3} \right) f \left( \frac{1}{3}, \frac{1}{3} \right) + H(-1) f(1, 1) \right] \]

\[ = \left( \frac{2}{3} \right)^2 f \left( \frac{1}{3}, \frac{1}{3} \right) \]
\[ T_0^2 = R_{x_1}^{[5/2,0]}[0, 1] R_{x_2}^{[5/2,0]}[0, 1 - x_1] f(x_1, x_2) \]
\[ = R_{x_1}^{[5/2,0]}[0, 1] \left[ \frac{1}{5/2} \sum_{j=-\infty}^{\infty} \left[H\left(1 - x_1 - \frac{2j - 1}{2 \cdot 5/2}\right) - H\left(\frac{1 - 2j}{2 \cdot 5/2}\right)\right] f\left(x_1, \frac{2j - 1}{2 \cdot 5/2}\right) \right] \]
\[ = R_{x_1}^{[5/2,0]}[0, 1] \left[ \frac{2}{5} \sum_{j=-\infty}^{\infty} \left[H\left(\frac{6 - 2j}{5} - x_1\right) - H\left(\frac{1 - 2j}{5}\right)\right] f\left(x_1, \frac{2j - 1}{5}\right) \right] \]
\[ = R_{x_1}^{[5/2,0]}[0, 1] \left[ \frac{2}{5} \left[ H\left(\frac{4}{5} - x_1\right) f\left(x_1, \frac{1}{5}\right) + H\left(\frac{2}{5} - x_1\right) f\left(x_1, \frac{3}{5}\right) \right] + H\left(\frac{0}{5} - x_1\right) f(x_1, 1) \right] \]
\[ = R_{x_1}^{[5/2,0]}[0, 1] g(x_1) \]
\[ = \frac{2}{5} \sum_{j=-\infty}^{\infty} \left[H\left(\frac{6 - 2j}{5}\right) - H\left(\frac{1 - 2j}{5}\right)\right] g\left(\frac{2j - 1}{5}\right) \]
\[ = \frac{2}{5} \left[ H\left(\frac{4}{5}\right) g\left(\frac{1}{5}\right) + H\left(\frac{2}{5}\right) g\left(\frac{3}{5}\right) + H(0) g(1) \right] \]
\[ = \left(\frac{2}{5}\right)^2 \left[ f\left(\frac{1}{5}, \frac{1}{5}\right) + \sum f\left(\frac{1}{5}, \frac{3}{5}\right) \right] \]

\[ T_0^3 = R_{x_1}^{[7/2,0]}[0, 1] R_{x_2}^{[7/2,0]}[0, 1 - x_1] f(x_1, x_2) \]
\[ = R_{x_1}^{[7/2,0]}[0, 1] \left[ \frac{2}{7} \sum_{j=-\infty}^{\infty} \left[H\left(\frac{8 - 2j}{7} - x_1\right) - H\left(\frac{1 - 2j}{7}\right)\right] f\left(x_1, \frac{2j - 1}{7}\right) \right] \]
\[ = R_{x_1}^{[7/2,0]}[0, 1] \left[ \frac{2}{7} \left[ H\left(\frac{6}{7} - x_1\right) f\left(x_1, \frac{1}{7}\right) + H\left(\frac{4}{7} - x_1\right) f\left(x_1, \frac{3}{7}\right) \right] + H\left(\frac{2}{7} - x_1\right) f(x_1, 1) \right] \]
\[ = R_{x_1}^{[7/2,0]}[0, 1] g(x_1) \]
\[ = \frac{2}{7} \sum_{j=-\infty}^{\infty} \left[H\left(\frac{8 - 2j}{7}\right) - H\left(\frac{1 - 2j}{7}\right)\right] g\left(\frac{2j - 1}{7}\right) \]
\[ = \frac{2}{7} \left[ H\left(\frac{6}{7}\right) g\left(\frac{1}{7}\right) + H\left(\frac{4}{7}\right) g\left(\frac{3}{7}\right) + H\left(\frac{2}{7}\right) g\left(\frac{5}{7}\right) + H(0) g(1) \right] \]
\[ = \left(\frac{2}{7}\right)^2 \left[ f\left(\frac{1}{7}, \frac{1}{7}\right) + \sum f\left(\frac{1}{7}, \frac{3}{7}\right) + \sum f\left(\frac{1}{7}, \frac{5}{7}\right) \right] \]
\( T_0^1 = R_{x_1}^{[0/2.0]} [0,1] R_{x_2}^{[0/2.0]} [0,1 - x_1] f(x_1, x_2) \)
\[ = \frac{2}{9} \sum_{j=-\infty}^{\infty} \left[ H \left( \frac{10 - 2j}{9} - x_1 \right) - H \left( \frac{1 - 2j}{9} \right) \right] g \left( \frac{2j - 1}{9} \right) \]
\[ = \frac{2}{9} \left[ H \left( \frac{8}{9} - x_1 \right) g \left( \frac{1}{9} \right) + H \left( \frac{6}{9} - x_1 \right) g \left( \frac{3}{9} \right) + H \left( \frac{4}{9} - x_1 \right) g \left( \frac{5}{9} \right) + H \left( \frac{2}{9} - x_1 \right) g \left( \frac{7}{9} \right) + H \left( 0 \right) g \left( 1 \right) \right] \]
\[ = \left( \frac{2}{9} \right)^2 \left[ f \left( \frac{1}{9} \right) + f \left( \frac{3}{9} \right) s + \sum f \left( \frac{1}{9} \frac{3}{9} \right) s + \sum f \left( \frac{1}{9} \frac{5}{9} \right) s \right] \]
\[ + \sum f \left( \frac{1}{9} \frac{7}{9} \right) s + \sum f \left( \frac{3}{9} \frac{5}{9} \right) s \]

\( T_0^2 = \frac{1}{2} \left( \frac{1}{3} \frac{1}{3} \right) \)
\( T_0^3 = \frac{25}{96} \left[ f \left( \frac{1}{5} \frac{1}{5} \right) \sum f \left( \frac{1}{5} \frac{3}{5} \right) s \right] - \frac{27}{96} f \left( \frac{1}{3} \frac{1}{3} \right) \)
\( T_0^4 = \frac{2401}{11520} \left[ f \left( \frac{1}{7} \frac{1}{7} \right) + f \left( \frac{3}{7} \frac{3}{7} \right) + \sum f \left( \frac{1}{7} \frac{3}{7} \right) s + \sum f \left( \frac{1}{7} \frac{5}{7} \right) s \right] \)
\[ - \frac{625}{2304} \left[ f \left( \frac{1}{5} \frac{1}{5} \right) + \sum f \left( \frac{1}{5} \frac{3}{5} \right) s \right] + \frac{81}{1280} f \left( \frac{1}{3} \frac{1}{3} \right) \]
\( T_0^5 = \frac{59049}{286720} \left[ f \left( \frac{1}{9} \frac{1}{9} \right) + f \left( \frac{3}{9} \frac{3}{9} \right) s + \sum f \left( \frac{1}{9} \frac{1}{9} \right) + \sum f \left( \frac{5}{9} \frac{1}{9} \right) s \right] \)
\[ + \sum f \left( \frac{1}{9} \frac{7}{9} \right) s + \sum f \left( \frac{3}{9} \frac{5}{9} \right) s \]
\[ - \frac{117649}{368640} \left[ f \left( \frac{1}{7} \frac{1}{7} \right) + f \left( \frac{3}{7} \frac{3}{7} \right) + \sum f \left( \frac{1}{7} \frac{3}{7} \right) s + f \left( \frac{1}{7} \frac{5}{7} \right) s \right] \]
\[ + \frac{15625}{129024} \left[ f \left( \frac{1}{5} \frac{1}{5} \right) + \sum f \left( \frac{1}{5} \frac{3}{5} \right) s \right] + \frac{81}{10240} f \left( \frac{1}{3} \frac{1}{3} \right) \]
Three dimensional Rules

\[
T_0^3 = \frac{1}{8} f \left( \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right)
\]

\[
T_0^2 \quad = \quad \left( \frac{1}{3} \right)^3 \left[ f \left( \frac{1}{6}, \frac{1}{6}, \frac{1}{6} \right) + \sum f \left( \frac{3}{6}, \frac{1}{6}, \frac{1}{6} \right) \right]
\]

\[
T_0^3 \quad = \quad \left( \frac{1}{4} \right)^3 \left[ f \left( \frac{1}{8}, \frac{1}{8}, \frac{1}{8} \right) + \sum f \left( \frac{3}{8}, \frac{1}{8}, \frac{1}{8} \right) + \sum f \left( \frac{5}{8}, \frac{1}{8}, \frac{1}{8} \right) + \sum f \left( \frac{3}{8}, \frac{3}{8}, \frac{1}{8} \right) \right]
\]

\[
T_0^4 \quad = \quad \left( \frac{1}{5} \right)^3 \left[ f \left( \frac{1}{10}, \frac{1}{10}, \frac{1}{10} \right) + \sum f \left( \frac{3}{10}, \frac{1}{10}, \frac{1}{10} \right) + \sum f \left( \frac{5}{10}, \frac{1}{10}, \frac{1}{10} \right) + \sum f \left( \frac{3}{10}, \frac{3}{10}, \frac{1}{10} \right) + \sum f \left( \frac{5}{10}, \frac{3}{10}, \frac{1}{10} \right) \right]
\]

\[
T_1^0 \quad = \quad \frac{1}{6} f \left( \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right)
\]

\[
T_2^0 \quad = \quad \frac{3}{40} \left[ f \left( \frac{1}{6}, \frac{1}{6}, \frac{1}{6} \right) + \sum f \left( \frac{3}{6}, \frac{1}{6}, \frac{1}{6} \right) \right] - \frac{2}{15} f \left( \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right)
\]

\[
T_3^0 \quad = \quad \frac{16}{315} \left[ f \left( \frac{1}{8}, \frac{1}{8}, \frac{1}{8} \right) + \sum f \left( \frac{3}{8}, \frac{1}{8}, \frac{1}{8} \right) + \sum f \left( \frac{5}{8}, \frac{1}{8}, \frac{1}{8} \right) + \sum f \left( \frac{3}{8}, \frac{3}{8}, \frac{1}{8} \right) \right] - \frac{27}{280} \left[ f \left( \frac{1}{6}, \frac{1}{6}, \frac{1}{6} \right) + \sum f \left( \frac{3}{6}, \frac{1}{6}, \frac{1}{6} \right) \right] + \frac{2}{45} f \left( \frac{1}{4}, \frac{1}{4}, \frac{1}{4} \right)
\]

\[
T_4^0 \quad = \quad \frac{3125}{72576} \left[ f \left( \frac{1}{10}, \frac{1}{10}, \frac{1}{10} \right) + \sum f \left( \frac{3}{10}, \frac{1}{10}, \frac{1}{10} \right) + \sum f \left( \frac{5}{10}, \frac{1}{10}, \frac{1}{10} \right) \right] + \sum f \left( \frac{7}{10}, \frac{1}{10}, \frac{1}{10} \right) + \sum f \left( \frac{3}{10}, \frac{3}{10}, \frac{1}{10} \right) + \sum f \left( \frac{5}{10}, \frac{3}{10}, \frac{1}{10} \right) \right]
\]
Four Dimensional Rules

\[
\begin{aligned}
T_0^4 &= 0 \\
T_1^4 &= 0 \\
T_2^4 &= \left(\frac{2}{5}\right)^4 \left[ f\left(\frac{1}{5}; \frac{1}{5}; \frac{1}{5}; \frac{1}{5}\right) \right] \\
T_3^4 &= \left(\frac{2}{7}\right)^4 \left[ f\left(\frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}\right) + \sum f\left(\frac{3}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}\right)_s \right] \\
T_4^4 &= \left(\frac{2}{9}\right)^4 \left[ f\left(\frac{1}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right) + \sum f\left(\frac{3}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right)_s \\
&\quad + \sum f\left(\frac{5}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right)_s + \sum f\left(\frac{3}{9}; \frac{3}{9}; \frac{1}{9}; \frac{1}{9}\right)_s \right] \\
T_5^4 &= \left(\frac{2}{11}\right)^4 \left[ f\left(\frac{1}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right) + \sum f\left(\frac{3}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right)_s + \sum f\left(\frac{5}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right)_s \\
&\quad + \sum f\left(\frac{7}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right)_s + \sum f\left(\frac{5}{11}; \frac{3}{11}; \frac{1}{11}; \frac{1}{11}\right)_s \\
&\quad + \sum f\left(\frac{3}{11}; \frac{3}{11}; \frac{1}{11}; \frac{1}{11}\right)_s + \sum f\left(\frac{3}{11}; \frac{3}{11}; \frac{3}{11}; \frac{1}{11}\right)_s \right] \\
T_0^5 &= 0 \\
T_1^5 &= \frac{1}{24} f\left(\frac{1}{5}; \frac{1}{5}; \frac{1}{5}; \frac{1}{5}; \frac{1}{5}\right) \\
T_2^5 &= \frac{49}{2880} \left[ f\left(\frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}\right) + \sum f\left(\frac{3}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}\right)_s \right] - \frac{25}{576} f\left(\frac{1}{5}; \frac{1}{5}; \frac{1}{5}; \frac{1}{5}; \frac{1}{5}\right) \\
T_3^5 &= \frac{729}{71680} \left[ f\left(\frac{1}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right) + \sum f\left(\frac{3}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right)_s + \sum f\left(\frac{5}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right)_s \\
&\quad + \sum f\left(\frac{3}{9}; \frac{3}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right)_s \right] - \frac{2401}{92160} \left[ f\left(\frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}\right) + \sum f\left(\frac{3}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}\right)_s \right] \\
&\quad + \frac{625}{32256} f\left(\frac{1}{5}; \frac{1}{5}; \frac{1}{5}; \frac{1}{5}; \frac{1}{5}\right) \\
T_4^5 &= \frac{1771561}{232243200} \left[ f\left(\frac{1}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right) + \sum f\left(\frac{3}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right)_s \\
&\quad + \sum f\left(\frac{5}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right)_s + \sum f\left(\frac{7}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right)_s + \sum f\left(\frac{5}{11}; \frac{3}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right)_s \\
&\quad + \sum f\left(\frac{3}{11}; \frac{3}{11}; \frac{1}{11}; \frac{1}{11}; \frac{1}{11}\right)_s + \sum f\left(\frac{3}{11}; \frac{3}{11}; \frac{3}{11}; \frac{1}{11}; \frac{1}{11}\right)_s \right] \\
&\quad - \frac{59049}{2867200} \left[ f\left(\frac{1}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right) + \sum f\left(\frac{3}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right)_s \\
&\quad + \sum f\left(\frac{5}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right)_s + \sum f\left(\frac{3}{9}; \frac{3}{9}; \frac{1}{9}; \frac{1}{9}; \frac{1}{9}\right)_s \right] \\
&\quad + \frac{117649}{6635520} \left[ f\left(\frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}\right) + \sum f\left(\frac{3}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}; \frac{1}{7}\right)_s \right] - \frac{15625}{3096576} f\left(\frac{1}{5}; \frac{1}{5}; \frac{1}{5}; \frac{1}{5}; \frac{1}{5}\right)
\end{aligned}
\]
We also derive a cubature formula that is exact for functions of $P_4^4$ in $T_4$ by using a set of prescribed abscissas.

The chosen basis of $P_4^4$ is

$$\{ 1, x_1, x_2, x_3, x_4, x_1 x_2, x_1 x_3, x_1 x_4, x_2 x_3, x_2 x_4, x_3 x_4, x_1^2, x_2^2, x_3^2, x_4^2 \}.$$

We choose the prescribed abscissas as

$$X = \left\{ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 1/2 & 1/2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1/2 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 1/2 \\ 1/2 & 1/2 & 1/2 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 1/2 & 1/2 & 1/2 \end{bmatrix} \right\}.$$ 

Then the set of moment equations expressed in matrix form is

$$\text{vol}(T_4) \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 1/2 & 1/2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1/2 & 0 & 0 & 1/2 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1/2 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \left[ \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \\ c_{10} \\ c_{11} \\ c_{12} \\ c_{13} \\ c_{14} \\ c_{15} \end{bmatrix} \right] = \int_{T_4} \begin{bmatrix} 1 & x_1 & x_2 & x_3 & x_4 & x_1 x_2 & x_1 x_3 & x_1 x_4 & x_2 x_3 & x_2 x_4 & x_3 x_4 & x_1^2 & x_2^2 & x_3^2 & x_4^2 \end{bmatrix}^T \, dx.$$
\[
\begin{bmatrix}
    c_1 \\
    c_2 \\
    c_3 \\
    c_4 \\
    c_5 \\
    c_6 \\
    c_7 \\
    c_8 \\
    c_9 \\
    c_{10} \\
    c_{11} \\
    c_{12} \\
    c_{13} \\
    c_{14} \\
    c_{15}
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
    \frac{-1}{15} \\
    \frac{-1}{15} \\
    \frac{-1}{15} \\
    \frac{-1}{15} \\
    \frac{2}{15} \\
    \frac{2}{15} \\
    \frac{2}{15} \\
    \frac{2}{15} \\
    \frac{2}{15} \\
    \frac{2}{15} \\
    \frac{2}{15} \\
    \frac{2}{15} \\
    \frac{2}{15} \\
    \frac{2}{15} \\
\end{bmatrix}
\]