On the Accuracy of Loader's Algorithm for the Binomial Density and Algorithms for Rectangle Probabilities for Markov Increments

Dissertation

zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften

Dem Fachbereich IV der Universität Trier
vorgelegt von

Jannis Dimitriadis

Trier, im August 2016

Betreuer: Prof. Dr. Lutz Mattner

Berichterstatter: Prof. Dr. Norbert Müller

Acknowledgement

I would like to thank Prof. Dr. Lutz Mattner for giving me the opportunity to work on this thesis and for very helpful advices during the work. Without his patience and continuous support this thesis would not have been completed. Prof. Dr. Lutz Mattner already was supervisor of my diploma thesis and a teacher of mine for many years. I learned very much from him.

I would like to thank Prof. Dr. Norbert Müller for agreeing to referee this thesis.

I would like to thank my family and my friends for supporting me during the years in which I worked on this thesis.

Zusammenfassung

In dieser Arbeit wird in Verallgemeinerung von Corrado [4] ein Algorithmus hergeleitet, welcher die Berechnung von Rechteckwahrscheinlichkeiten für Markov-Inkremente ermöglicht. Es wird gezeigt, daß es sich bei multinomialverteilten und bei multivariat hypergeometrisch verteilten Zufallsgrößen um Markov-Inkremente handelt. In einem Beispiel wird gezeigt, daß der hergeleitete Algorithmus im Multinomialfall schneller ein Ergebnis liefert, als eine herkömmliche Methode, bei welcher alle Elemente des Trägers der Multinomialverteilung konstruiert werden und deren relevante Einpunktwahrscheinlichkeiten aufsummiert. Als Anwendung des hergeleiteten Algorithmus wird eine Verteilung der Spannweite einer multinomial verteilten Zufallsgrößen berechnet. Für die Untersuchung der Rechengenauigkeit bei dem hergeleiteten Algorithmus ist es im Multinomialfall nötig, zunächst die Genauigkeit eines Algorithmus zu untersuchen, welcher Einpunktwahrscheinlichkeiten von Binomialverteilungen berechnet. Dies geschieht bei dem Statistik Softwarepaket R mit einem Algorithmus nach Loader [16]. Daher werden Hilfsresultate hergeleitet, welche dazu dienen können, einen Satz über die Rechengenauigkeit des Algorithmus nach Loader herzuleiten. Zudem werden in Beispielen die Genauigkeit des hergeleiteten Algorithmus im Multinomialfall sowie im multivariat hypergeometrischen Fall untersucht mit Hilfe von intervallarithmetischen Berechnungen. Es wird folgendes statistische Anwendungsbeispiel untersucht: Es kommen n Patienten in einer Klinik an $d = 365$ Tagen des Jahres an, jeder der Patienten mit Wahrscheinlichkeit $1/d$ an jedem dieser d Tage und alle Patienten unabhängig voneinander. Wie groß ist die Wahrscheinlichkeit, daß 3 aufeinanderfolgende Tage existieren, an denen zusammen mehr als k Patienten ankommen?

Contents

Introduction

The main achievement of this thesis is an analysis of the accuracy of computations with Loader's algorithm for the binomial density. This analysis in later progress of work could be used for a theorem about the numerical accuracy of algorithms that compute rectangle probabilities for scan statistics of a multinomially distributed random variable. An example that shall illustrate the practical use of probabilities for scan statistics is the following, which arises in epidemiology: Let n patients arrive at a clinic in $d = 365$ days, each of the patients with probability $1/d$ at each of these d days and all patients independently from each other. The knowledge of the probability, that there exist 3 adjacent days, in which together more than k patients arrive, helps deciding, after observing data, if there is a cluster which we would not suspect to have occurred randomly but for which we suspect there must be a reason. Formally, this epidemiological example can be described by a multinomial model. As multinomially distributed random variables are examples of Markov increments, which is a fact already used implicitly by Corrado [4] to compute the distribution function of the multinomial maximum, we can use a generalized version of Corrado's Algorithm to compute the probability described in our example. To compute its result, the algorithm for rectangle probabilities for Markov increments always uses transition probabilities of the corresponding Markov Chain. In the multinomial case, the transition probabilities of the corresponding Markov Chain are binomial probabilities. Therefore, we start an analysis of accuracy of Loader's algorithm for the binomial density, which for example the statistical software `R` [20] uses. With the help of accuracy bounds for the binomial density we would be able to derive accuracy bounds for the computation of rectangle probabilities for scan statistics of multinomially distributed random variables. To figure out how sharp derived accuracy bounds are, in examples these can be compared to rigorous upper bounds and rigorous lower bounds which we obtain by interval-arithmetical computations.

Chapter 1

Algorithms for the computation of rectangle probablilities for Markov increments

1.1 Proof of correctness for an abstract algorithm for the computation of rectangle probablilities for Markov increments

In this section we describe an abstract algorithm for the computation of rectangle probabilities for Markov increments. We prove that if the operations $+, \cdot$ on $\mathbb{R}$ are performed exactly, this algorithm is correct.

Definition 1.1. Let $(\mathcal{X}, \cdot)$ be a group with $\mathcal{X}$ a countable set and $(X_k)_{k=1}^d$ a Markov chain on the probability space $(\Omega, \mathcal{A}, \mathbb{P})$ which takes values in the measurable space $(\mathcal{X}, 2^{\mathcal{X}})$. Let $Y_1 := X_1$ and $Y_k := X_{k-1}^{-1} X_k$ for $k \in \{2, \ldots, d\}$. Then the family $(Y_k)_{k=1}^d$ is called the **(Markov) increment** of the Markov chain $(X_k)_{k=1}^d$.

We remark that if $(Y_k)_{k=1}^d$ is the increment of the Markov chain $(X_k)_{k=1}^d$, then we have

$$X_k = Y_1 \cdot \ldots \cdot Y_k$$

for $k \in \{1, \ldots, d\}$.

Corrado [4] uses the algorithm A, which we state in this section below, for the computation of rectangle probabilities for Markov increments. It is based on the recursion formula (1.1) we state in the following theorem.

Theorem 1.2. *Let $Y = (Y_k)_{k=1}^d$ be Markov increment of a Markov chain $(X_k)_{k=1}^d$ which takes values in a group $(\mathcal{X}, \cdot)$. Let $A_1, \ldots, A_d \subseteq \mathcal{X}$ be countable sets. Then the probabilities*

$$p(k, x) := \mathbb{P}(X_k = x, Y_1 \in A_1, \ldots, Y_k \in A_k)$$

for $k \in \{1, \ldots, d\}$ and $x \in \mathcal{X}$ fulfill the recursion

$$p(k,x) = \sum_{y \in A_k} \mathbb{P}(X_k = x \mid X_{k-1} = xy^{-1}) p(k-1, xy^{-1}) \tag{1.1}$$

for $k \geq 2$. Here and throughout, we use the convention $\mathbb{P}(A|B) = \mathbb{P}(A \cap B)/\mathbb{P}(B) := 0$ if $\mathbb{P}(B) = 0$.

Proof. The functions $f_k : \mathcal{X}^2 \to \mathcal{X}$ defined by $f_k(x_1, x_2) = x_1^{-1} x_2$ have the property that $Y_k = f_k(X_{k-1}, X_k)$ and $f_k(\cdot, x)$ is bijective for every $x \in \mathcal{X}$. Using this (which is actually all we need, so the method works not only for Markov increments but actually for any functions of two successive states of a Markov chain having the above bijectivity property) and writing $g_k(x, \cdot) := f_k(\cdot, x)^{-1}$, we get:

$$\begin{aligned}
&\mathbb{P}(X_k = x, Y_1 \in A_1, \ldots, Y_k \in A_k) \\
&= \sum_{y \in A_k} \mathbb{P}(X_k = x, Y_k = y, Y_1 \in A_1, \ldots, Y_{k-1} \in A_{k-1}) \\
&= \sum_{y \in A_k} \mathbb{P}(X_k = x, X_{k-1} = g_k(x,y), Y_1 \in A_1, \ldots, Y_{k-1} \in A_{k-1}) \\
&= \sum_{y \in A_k} \mathbb{P}(X_k = x | X_{k-1} = g_k(x,y)) \\
&\qquad \times \mathbb{P}(X_{k-1} = g_k(x,y), Y_1 \in A_1, \ldots, Y_{k-1} \in A_{k-1})
\end{aligned}$$

In the last step the Markov property was used. □

From the recursion formula we can derive the following algorithm that computes the probability $\mathbb{P}(Y_1 \in A_1, \ldots, Y_d \in A_d)$. We assume that $A_1, \ldots, A_d$ are finite, so we get a finite algorithm.

Algorithm A:

1. For every $x \in A_1$ compute the value $p(1,x) = \mathbb{P}(X_1 = x)$
2. For every $k \in \{2, \ldots, d\}$:
   For every $x \in A_1 \cdot \ldots \cdot A_k$ compute the value $p(k,x)$ with formula (1.1)
3. Compute $\mathbb{P}(Y_1 \in A_1, \ldots, Y_d \in A_d) = \sum_{x \in A_1 \cdot \ldots \cdot A_d} p(d,x)$

Here, let $A_1 \cdots A_n := \{a_1 \cdots a_n : a_1 \in A_1, \ldots, a_n \in A_n\}$, if $\mathcal{X}$ is a group and $A_1, \ldots, A_n \subset \mathcal{X}$.

Now, we describe how to compute a rectangle scan probability

$$q := P(Y_1 \cdot \ldots \cdot Y_\ell \in A_1, \ldots, Y_{d-\ell+1} \cdot \ldots \cdot Y_d \in A_{d-\ell+1})$$

for a Markov increment Y.

We use the following obvious and well-known lemma:

Lemma 1.3. *Let $\mathcal{X}$ be a countable set and $(X_k)_{k=1}^d$ an $\mathcal{X}$-valued Markov chain. Let $W_k := (X_k, \ldots, X_{k+\ell-1})$. Then $(W_k)_{k=1}^{d-\ell+1}$ is an $\mathcal{X}^\ell$-valued Markov chain with transition probabilities*

$$\mathbb{P}(W_{k+1} = w \mid W_k = v) = \mathbb{P}(X_{k+\ell} = w_\ell \mid X_{k+\ell-1} = v_\ell)$$

for $v, w \in \mathcal{X}^\ell$ with $\mathbb{P}(W_k = v) > 0$ and $v_2 = w_1, \ldots, v_\ell = w_{\ell-1}$.

The desired rectangle scan probability for the Markov increment Y can be written as a rectangle probability for the increment V of W: If we set $B_k := \{(y_1, \ldots, y_\ell) \in \mathcal{X}^\ell | y_1 \cdot \ldots \cdot y_\ell \in A_k\}$ we have

$$q = \mathbb{P}(V_1 \in B_1, \ldots, V_{d-\ell+1} \in B_{d-\ell+1})$$

because $V_k = (X_{k-1}^{-1} X_k, \ldots, X_{k+\ell}^{-1} X_{k+\ell-1})$ for $k \in \{2, \ldots, d-\ell+1\}$.

The sets $B_1, \ldots, B_{d-\ell+1}$ are possibly infinite so the Algorithm A would not compute a result in finite time. But if there exist finite sets $M_1, \ldots, M_{d-\ell+1} \subseteq \mathcal{X}^\ell$ with

$$\mathbb{P}(V_1 \in B_1, \ldots, V_{d-\ell+1} \in B_{d-\ell+1}) = \mathbb{P}(V_1 \in M_1, \ldots, V_{d-\ell+1} \in M_{d-\ell+1})$$

we can apply the Algorithm A and thus are able to compute the desired probability.

Example: If $\mathcal{X} = (\mathbb{Z}, +)$ and Y is a Markov increment with $Y_1, \ldots, Y_d \geq 0$, then for finite sets $A_1, \ldots, A_{d-\ell+1} \subseteq \mathbb{Z}$ the probability

$$\mathbb{P}(Y_1 + \ldots + Y_\ell \in A_1, \ldots, Y_{d-\ell+1} + \ldots + Y_d \in A_d)$$

equals

$$\mathbb{P}((Y_1, \ldots, Y_\ell) \in M_1, \ldots, (Y_{d-\ell+1}, \ldots, Y_d) \in M_{d-\ell+1})$$

with $M_k := \{(y_1, \ldots, y_\ell) \in \mathbb{N}_0^\ell | y_1 + \ldots + y_\ell \in A_k\}$, which are finite.

In the next section we will prove that multinomially distributed random variables are Markov increments and the transition probabilities of the corresponding Markov chain are binomial probabilities. Therefore the Algorithm A can be used to compute rectangle scan probabilities for multinomially distributed random variables.

1.2 Multinomially distributed random variables are Markov increments

We define the multinomial distribution and the binomial distribution by their densities.

Definition 1.4. (a) Let $n, d \in \mathbb{N}$ with $d \geq 2$ and $p \in [0,1]^d$ with $\sum_{i=1}^d p_i = 1$. The **multinomial distribution** $\mathrm{M}_{n,p}$ is the probability measure on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ with

$$\mathrm{M}_{n,p}(\{(n_1, \ldots, n_d)\}) = \binom{n}{n_1, \ldots, n_d} p_1^{n_1} \cdot \ldots \cdot p_d^{n_d}$$

for $n_1, \ldots, n_d \in \mathbb{N}_0$ with $n_1 + \ldots + n_d = n$.
(b) Let $n \in \mathbb{N}$ and $p \in [0,1]$. The **binomial distribution** $\mathrm{B}_{n,p}$ is the probability measure on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ with

$$\mathrm{B}_{n,p}(\{x\}) = \mathrm{b}_{n,p}(x) := \binom{n}{x} p^x (1-p)^{n-x}$$

for $x \in \{0, \ldots, n\}$. For $n = 0$ let $\mathrm{B}_{n,p} := \delta_0$ be the Dirac measure in 0 and $\mathrm{b}_{n,p}$ its density.

Our aim in this section is to show that multinomially distributed random variables are Markov increments. The derivation in this section follows [5] and we use the fact that the multinomial distribution $\mathrm{M}_{n,p}$ is the distribution of the sum $\sum_{i=1}^n X_i$ of n independent random variables $X_1, \ldots, X_n$, each with d-dimensional Bernoulli distribution $\mathrm{B}_p := \sum_{i=1}^d p_i \delta_{e_i}$. Here $e_i := ((i = j))_{j \in \{1,\ldots,d\}}$ denotes the i-th unit vector in $\mathbb{R}^d$.

We begin with an easy lemma on the special case of a 2-dimensional multinomially distributed random variable, which can be written as $(X, n - X)$ with a binomially distributed random variable X.

Lemma 1.5. *Let $n \in \mathbb{N}$, $p \in [0;1]$ and $X : \Omega \to \mathbb{R}$ a random variable on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Then we have*

$$(X, n - X) \sim \mathrm{M}_{n,(p,1-p)} \iff X \sim \mathrm{B}_{n,p}$$

Proof. We have

$$\begin{aligned}
(X, n - X) \sim \mathrm{M}_{n,(p,1-p)} &\iff \mathbb{P}(X = x, n - X = y) = \binom{n}{x, y} p^x (1-p)^y \text{ for } x, y \in \mathbb{N}_0 \\
&\iff \mathbb{P}(X = x) = \mathrm{B}_{n,p}(\{x\}) \text{ for } x \in \mathbb{N}_0 \\
&\iff X \sim \mathrm{B}_{n,p}
\end{aligned}$$

□

Condensed boxes

The multinomial model $\mathrm{M}_{n,p}$ with $p \in [0,1]^d, \sum_{i=1}^d p_i = 1$ can be illustrated by the idea of n balls independently falling into d boxes, each ball falling into box i with probability p_i. When different of these boxes in the multinomial model are condensed, again a multinomial model results. This is proposed in the next theorem.

Theorem 1.6. *Let $n, d \in \mathbb{N}$ with $d \geq 2$ and $p \in [0,1]^d$ with $\sum_{i=1}^d p_i = 1$. Let $N \sim \mathrm{M}_{n,p}$ a multinomially distributed random variable. Let $\ell \geq 2$ a natural number and $\{K_1, \ldots, K_\ell\}$ a partition of $\{1, \ldots, d\}$. Further we define for $r \in \{1, \ldots, \ell\}$ the random variable $U_r := \sum_{i \in K_r} N_i$ and $q_r := \sum_{i \in K_r} p_i$. Then we have $(U_1, \ldots, U_\ell) \sim \mathrm{M}_{n,(q_1,\ldots,q_\ell)}$.*

Proof. Let $X_1, \ldots, X_n \sim \mathrm{B}_p$ independent random variables with $N = \sum_{k=1}^n X_k$ and for $k \in \{1, \ldots, n\}$ and $r \in \{1, \ldots, \ell\}$ let $Y_{k,r} := \sum_{i \in K_r} X_{k,i}$. Then for $r \in \{1, \ldots, \ell\}$ we have

$$\sum_{k=1}^n Y_{k,r} = \sum_{k=1}^n \sum_{i \in K_r} X_{k,i} = \sum_{i \in K_r} \sum_{k=1}^n X_{k,i} = \sum_{i \in K_r} N_i = U_r$$

and therefore $(U_1, \ldots, U_\ell) = \sum_{k=1}^n (Y_{k,1}, \ldots, Y_{k,\ell})$. Because the familiy $((Y_{k,1}, \ldots, Y_{k,\ell}))_{k=1}^n$ is independent and each $(Y_{k,1}, \ldots, Y_{k,\ell}) \sim \mathrm{B}_{(q_1,\ldots,q_\ell)}$ Bernoulli distributed with the same parameter $(q_1, \ldots, q_\ell)$, we get $(U_1, \ldots, U_\ell) \sim \mathrm{M}_{n,(q_1,\ldots,q_\ell)}$. □

Marginal distributions

In the next Corollary we state the marginal distributions of a multinomially distributed random variable.

Corollary 1.7. *Let $n, d \in \mathbb{N}$ with $d \geq 2$ and $p \in [0,1]^d$ with $\sum_{i=1}^d p_i = 1$. Let $N \sim \mathrm{M}_{n,p}$ a multinomially distributed random variable. Let $k \in \{1, \ldots, d-1\}$ and $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, d\}$. Let $A_1, \ldots, A_k \subseteq \{0, \ldots, n\}$. Then*

$$\mathbb{P}(N_{i_1} \in A_1, \ldots, N_{i_k} \in A_k) = \mathrm{M}_{n,(p_{i_1},\ldots,p_{i_k},1-p_{i_1}-\ldots-p_{i_k})}(A_1 \times \ldots \times A_k \times \{0, \ldots, n\}) \tag{1.2}$$

Particularly we get that for $i \in \{1, \ldots, d\}$ the random variable N_i has the binomial distribution B_{n,p_i}.

Proof. From Theorem 1.6 we get $(N_{i_1}, \ldots, N_{i_k}, n - N_{i_1} - \ldots - N_{i_k}) \sim \mathrm{M}_{n,(p_{i_1},\ldots,p_{i_k},1-p_{i_1}-\ldots-p_{i_k})}$ and from this we get equation (1.2). Because of Lemma 1.5 this implies $N_i \sim \mathrm{B}_{n,p_i}$ for every $i \in \{1, \ldots, d\}$. □

Multinomially distributed random variables are Markov increments

Let $n, d \in \mathbb{N}$ with $d \geq 2$ and $p \in [0,1]^d$ with $\sum_{i=1}^d p_i = 1$. Let $N \sim \mathrm{M}_{n,p}$ a multinomially distributed random variable. We define

$$S_k := \sum_{i=1}^k N_i \qquad \text{for } k \in \{1, ..., d\}$$

Our aim in this section is Corollary 1.10. In that Corollary we will show that $(S_k)_{k=1}^d$ is a Markov chain. Therefore we get that the multinomially distributed random variable N is a Markov increment. For simplicity we assume $p \in]0;1[^d$.

Theorem 1.8. *Let $k \in \{2, ..., d\}$, $s_{k-1}, s_k \in \{0, ..., n\}$. Then we have*

$$\mathbb{P}(S_k = s_k | S_{k-1} = s_{k-1}) = b_{n-s_{k-1}, p_k^*}(s_k - s_{k-1})$$

with $p_k^* := p_k / \sum_{i=k}^d p_i$.

Proof. For $s_{k-1} > s_k$ the proposition is obviously true. Now let $s_{k-1} \leq s_k$ and $q := \sum_{i=1}^{k-1} p_i$. With Theorem 1.6 we get

$$\begin{aligned}
&\mathbb{P}(S_{k-1} = s_{k-1}, S_k = s_k) \\
&= P(S_{k-1} = s_{k-1}, N_k = s_k - s_{k-1}, n - S_k = n - s_k) \\
&= M_{n,(q,p_k,1-q-p_k)}(\{(s_{k-1}, s_k - s_{k-1}, n - s_k)\}) \\
&= \frac{n!}{s_{k-1}!(s_k - s_{k-1})!(n - s_k)!} q^{s_{k-1}} p_k^{s_k - s_{k-1}} (1 - q - p_k)^{n - s_k}
\end{aligned}$$

und from Theorem 1.6 and Corollary 1.7

$$\mathbb{P}(S_{k-1} = s_{k-1}) = b_{n,q}(s_{k-1}) = \frac{n!}{s_{k-1}!(n - s_{k-1})!} q^{s_{k-1}} (1 - q)^{n - s_{k-1}}$$

Altogether we get

$$\begin{aligned}
&\mathbb{P}(S_k = s_k | S_{k-1} = s_{k-1}) \\
&= \frac{P(S_k = s_k, S_{k-1} = s_{k-1})}{P(S_{k-1} = s_{k-1})} \\
&= \frac{(n - s_{k-1})!}{(s_k - s_{k-1})!(n - s_k)!} p_k^{s_k - s_{k-1}} (1 - q - p_k)^{n - s_k} / (1 - q)^{n - s_{k-1}} \\
&= \frac{(n - s_{k-1})!}{(s_k - s_{k-1})!(n - s_k)!} \left(\frac{p_k}{1 - q}\right)^{s_k - s_{k-1}} \left(1 - \frac{p_k}{1 - q}\right)^{n - s_k} \\
&= b_{n - s_{k-1}, p_k^*}(s_k - s_{k-1})
\end{aligned}$$

□

To show that $(S_k)_{k=1}^d$ is a Markov chain, we need the following theorem about the conditional distribution of the random variables $N_1, \ldots, N_k$, given $N_{k-1}, \ldots, N_1$.

Theorem 1.9. *Let $n_1, ..., n_d \in \{0, ..., n\}$ and $k \in \{2, ..., d\}$ with $\mathbb{P}(N_1 = n_1, \ldots, N_k = n_{k-1}) > 0$. Then with $m := \sum_{i=k}^d n_i$ and $q := \sum_{i=k}^d p_i$ we have*

$$\begin{aligned}
&\mathbb{P}(N_d = n_d, ..., N_k = n_k | N_{k-1} = n_{k-1}, ..., N_1 = n_1) \\
&\qquad = M_{m,(p_k/q, ..., p_d/q)}(\{(n_k, ..., n_d)\})
\end{aligned}$$

Proof. We have

$$\mathbb{P}(N_1 = n_1, ..., N_d = n_d) = M_{n,(p_1,...,p_d)}(\{(n_1, ..., n_d)\}) = \frac{n!}{n_1!...n_d!} p_1^{n_1} ... p_d^{n_d}$$

and because of Theorem 1.6

$$\begin{aligned} \mathbb{P}(N_1 = n_1, ..., N_{k-1} = n_{k-1}) &= M_{n,(p_1,...,p_{k-1},q)}(\{(n_1, ..., n_{k-1}, m)\}) \\ &= \frac{n!}{n_1!...n_{k-1}!m!} p_1^{n_1} ... p_{k-1}^{n_{k-1}} q^m \end{aligned}$$

Hence

$$\begin{aligned} &\mathbb{P}(N_d = n_d, ..., N_k = n_k | N_{k-1} = n_{k-1}, ..., N_1 = n_1) \\ &= \frac{P(N_1 = n_1, ..., N_d = n_d)}{P(N_1 = n_1, ..., N_{k-1} = n_{k-1})} \\ &= \frac{m!}{n_k!...n_d!} p_k^{n_k} ... p_d^{n_d} / q^m \\ &= \frac{m!}{n_k!...n_d!} (p_k/q)^{n_k} ... (p_d/q)^{n_d} \\ &= M_{m,(p_k/q,...,p_d/q)}(\{(n_k, ..., n_d)\}) \end{aligned}$$

□

Corollary 1.10. *The family $(S_k)_{k=1}^d$ is a Markov chain. For $k \in \{2, \dots, d\}$ and $s_{k-1}, s_k \in \{0, ..., n\}$ we have*

$$\mathbb{P}(S_k = s_k | S_{k-1} = s_{k-1}) = b_{n-s_{k-1},p_k^*}(s_k - s_{k-1})$$

with $p_k^* := p_k / \sum_{i=k}^d p_i$. Therefore the multinomially distributed random variable N is a Markov increment.

Proof. Let $k \in \{2, ..., d\}$ and $s_1, ..., s_k \in \{0, ..., n\}$ with $\mathbb{P}(S_1 = s_1, ..., S_{k-1} = s_{k-1}) > 0$. Then from Theorem 1.9 and Corollary 1.7 we get

$$\begin{aligned} &\mathbb{P}(S_k = s_k | S_{k-1} = s_{k-1}, ..., S_1 = s_1) \\ &= \frac{\mathbb{P}(S_k = s_k, ..., S_1 = s_1))}{\mathbb{P}(S_{k-1} = s_{k-1}, ..., S_1 = s_1)} \\ &= \frac{\mathbb{P}(N_k = s_k - s_{k-1}, \dots, N_2 = s_2 - s_1, N_1 = s_1))}{\mathbb{P}(N_{k-1} = s_{k-1} - s_{k-2}, \dots, N_2 = s_2 - s_1, N_1 = s_1)} \\ &= \mathbb{P}(N_k = s_k - s_{k-1} \mid N_{k-1} = s_{k-1} - s_{k-2}, \dots, N_2 = s_2 - s_1, N_1 = s_1) \\ &= b_{n-s_{k-1},p_k^*}(s_k - s_{k-1}) \end{aligned}$$

und therefore with Theorem 1.8 the proposition. □

We conclude that the Algorithm A can be used to compute rectangle scan probabilities

$$\mathbb{P}(N_1 + \ldots + N_\ell \leq k, \ldots, N_{d-\ell+1} + \ldots + N_d \leq k)$$

for a multinomially distributed random variable $N = (N_1, \ldots, N_d)$. This method is much faster than the enumerative method stated in the Appendix G. The enumerative method works as follows: Let $D = \{x \in \mathbb{N}_0^d : x_1 + \ldots + x_d = n\}$ the support of the multinomial distribution $\mathrm{M}_{n,p}$. For each $x \in D$ with $x_1 + \ldots + x_\ell \leq k, \ \ldots, \ x_{d-\ell+1} + \ldots + x_d \leq k$ compute the probability $\mathbb{P}(N = x) = n!/(x_1! \ldots x_d!) p_1^{x_1} \ldots p_d^{x_d}$ and sum up these values. But because the support D is large, this procedure takes much time. For example: For $n = 20, d = 12, p = (1/d, \ldots, 1/d)$ it took 41 minutes and 38 seconds with the enumerative algorithm stated in G to compute the probability $\mathbb{P}(N_1 + N_2 + N_3 \leq 9, \ldots, N_{d-2} + N_{d-1} + N_d \leq 9) = 0.88744$ on a 3.7 GHz CPU with 4.0 GB Ram, while with the implementation of Algorithm A which is stated in Appendix B it took less than 1 second.

1.3 Multivariate hypergeometrically distributed random variables are Markov increments

In this section we cover another important example for Markov increments, namely multivariate hypergeometrically distributed random variables. The multivariate hypergeometrical distribution is a model for drawing from an urn with balls of different colors, without replacing the drawn balls. Following the derivation in [5], in this section we show that multivariate hypergeometrically distributed random variables are Markov increments. Further properties of the multivariate hypergeometrical distribution can be found in [11].

Let $n, d \in \mathbb{N}, d \geq 2$ and $m_1, \ldots, m_d \in \mathbb{N}$ with $\sum_{i=1}^d m_i \geq n$.

Definition 1.11. (a) The **multivariate hypergeometrical distribution** $\mathrm{H}_{n,(m_1,\ldots,m_d)}$ is the probability measure on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ with

$$\mathrm{H}_{n,(m_1,\ldots,m_d)}(\{k_1, \ldots, k_d\}) = \frac{\binom{m_1}{k_1} \cdot \ldots \cdot \binom{m_d}{k_d}}{\binom{\sum_{i=1}^d m_i}{n}}$$

for $(k_1, \ldots, k_d) \in \{0, \ldots, m_1\} \times \ldots \times \{0, \ldots, m_d\}$ with $k_1 + \ldots + k_d = n$.

(b) Let $r, b \in \mathbb{N}_0$ with $r + b \geq n$. The **hypergeometrical distribution** $\mathrm{H}_{n,r,b}$ is the probability measure on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ with

$$\mathrm{H}_{n,r,b}(\{k\}) = \mathrm{h}_{n,r,b}(k) := \frac{\binom{r}{k}\binom{b}{n-k}}{\binom{r+b}{n}}$$

for $k \in \{0, \ldots, n\}$. In case of $n = 0$ let $\mathrm{H}_{n,r,b} = \delta_0$ the Dirac measure in 0 and $\mathrm{h}_{n,r,b}$ its density.

In the rest of this section let $N = (N_1, ..., N_d)$ be a $\mathrm{H}_{n,(m_1,...,m_d)}$-distributed random variable and $m := \sum_{i=1}^{d} m_i$.

We need the following theorem.

Theorem 1.12. *Let $\ell \in \mathbb{N}, \ell \geq 2, \{T_1, ..., T_\ell\}$ a partition of $\{1, ..., d\}$ and $S_r := \sum_{i \in T_r} N_i$ and $s_r := \sum_{i \in T_r} m_i$ for $r \in \{0, ..., \ell\}$. Then the random variable $(S_1, ..., S_\ell)$ has distribution $\mathrm{H}_{n,(s_1,...,s_\ell)}$.*

Proof. This Theorem can be proven with the help of the Vandermonde convolution identity, which for $k \in \mathbb{N}_0$ reads as follows

$$\sum_{\substack{(k_1,...,k_d) \in \mathbb{N}_0^d \\ k_1+...+k_d=k}} \binom{m_1}{k_1} \cdots \binom{m_d}{k_d} = \binom{m_1 + \ldots + m_d}{k}$$

□

Corollary 1.13. *For $i \in \{1, ..., d\}$ we have*

$$N_i \sim \mathrm{H}_{n,m_i,m-m_i}$$

Proof. For $i \in \{1, ..., d\}$ from Theorem 1.12 we get that the random variable $(N_i, n - N_i)$ has distribution $\mathrm{H}_{n,(m_i,m-m_i)}$. From this we get the proposition. □

We define

$$S_k := \sum_{i=1}^{k} N_i \text{ for } k \in \{1, ..., d\}$$

Theorem 1.14. *Let $k \in \{1, ..., d\}$ and $s_{k-1}, s_k \in \{0, ..., n\}$. Then we have*

$$\mathbb{P}(S_k = s_k | S_{k-1} = s_{k-1}) = \mathrm{h}_{n-s_{k-1},m_k,m-s}(s_k - s_{k-1})$$

with $s := \sum_{i=1}^{k} m_i$.

Proof. For $s_{k-1} > s_k$ the proposition obviously is true. Now let $s_{k-1} \leq s_k$. From Theorem 1.12 we get

$$\begin{aligned}
&\mathbb{P}(S_{k-1} = s_{k-1}, S_k = s_k) \\
&= P(S_{k-1} = s_{k-1}, N_k = s_k - s_{k-1}, n - S_k = n - s_k) \\
&= \mathrm{H}_{n,(s-m_k,m_k,m-s)}(\{(s_{k-1}, s_k - s_{k-1}, n - s_k)\}) \\
&= \binom{s - m_k}{s_{k-1}} \binom{m_k}{s_k - s_{k-1}} \binom{m - s}{n - s_k} / \binom{m}{n}
\end{aligned}$$

From Theorem 1.12 and Corollary 1.13 we get

$$\mathbb{P}(S_{k-1} = s_{k-1}) = \mathrm{h}_{n,s-m_k,m-s+m_k}(s_{k-1}) = \binom{s-m_k}{s_{k-1}}\binom{m-s+m_k}{n-s_{k-1}}/\binom{m}{n}$$

Jointly we get

$$\begin{aligned}
&\mathbb{P}(S_k = s_k | S_{k-1} = s_{k-1}) \\
&= \frac{P(S_k = s_k, S_{k-1} = s_{k-1})}{P(S_{k-1} = s_{k-1})} \\
&= \binom{m_k}{s_k - s_{k-1}}\binom{m-s}{n-s_k}/\binom{m-s+m_k}{n-s_{k-1}} \\
&= \mathrm{h}_{n-s_{k-1},m_k,m-s}(s_k - s_{k-1})
\end{aligned}$$

□

Theorem 1.15. *Let* $n_1, ..., n_d \in \{0, ..., n\}$ *and* $k \in \{2, ..., d\}$ *with* $\mathbb{P}(N_1 = n_1, \ldots, N_{k-1} = n_{k-1}) > 0$. *Then with* $r := \sum_{i=k}^{d} n_i$ *we have*

$$\begin{aligned}
&\mathbb{P}(N_d = n_d, ..., N_k = n_k | N_{k-1} = n_{k-1}, ..., N_1 = n_1) \\
&\qquad = \mathrm{H}_{r,(m_k,...,m_d)}(\{(n_k, ..., n_d)\})
\end{aligned}$$

Proof. We have

$$\mathbb{P}(N_1 = n_1, ..., N_d = n_d) = \mathrm{H}_{n,(m_1,...,m_d)}(\{(n_1, ..., n_d)\}) = \binom{m_1}{n_1}...\binom{m_d}{n_d}/\binom{m}{n}$$

and because of Theorem 1.12

$$\begin{aligned}
\mathbb{P}(N_1 = n_1, ..., N_{k-1} = n_{k-1}) &= \mathrm{H}_{n,(m_1,...,m_{k-1},s)}(\{(n_1, ..., n_{k-1}, r)\}) \\
&= \binom{m_1}{n_1}...\binom{m_{k-1}}{n_{k-1}}\binom{s}{r}/\binom{m}{n}
\end{aligned}$$

with $s := \sum_{i=k}^{d} m_i$. This implies

$$\begin{aligned}
&\mathbb{P}(N_d = n_d, ..., N_k = n_k | N_{k-1} = n_{k-1}, ..., N_1 = n_1) \\
&= \frac{P(N_1 = n_1, ..., N_d = n_d)}{P(N_1 = n_1, ..., N_{k-1} = n_{k-1})} \\
&= \binom{m_k}{n_k}...\binom{m_d}{n_d}/\binom{s}{r} \\
&= \mathrm{H}_{r,(m_k,...,m_d)}(\{(n_k, ..., n_d)\})
\end{aligned}$$

□

Corollary 1.16. *The family $(S_k)_{k=1}^d$ is a Markov chain. For $k \in \{2,\dots,d\}$ and $s_{k-1}, s_k \in \{0,...,n\}$ we have*

$$\mathbb{P}(S_k = s_k | S_{k-1} = s_{k-1}) = \mathrm{h}_{n-s_{k-1},m_k,m-s}(s_k - s_{k-1})$$

with $s := \sum_{i=1}^k m_i$. Therefore, the multivariate hypergeometrically distributed random variable N is a Markov increment.

Proof. Let $k \in \{2,...,d\}$ and $s_1,...,s_k \in \{0,...,n\}$ with $\mathbb{P}(S_1 = s_1,\dots,S_{k-1} = s_{k-1}) > 0$ and $s := \sum_{i=1}^k m_i$. Then from Theorem 1.15 and Corollary 1.13 we get

$$\begin{aligned}
&\mathbb{P}(S_k = s_k | S_{k-1} = s_{k-1},...,S_1 = s_1) \\
&= \frac{\mathbb{P}(S_k = s_k,...,S_1 = s_1))}{\mathbb{P}(S_{k-1} = s_{k-1},...,S_1 = s_1)} \\
&= \frac{\mathbb{P}(N_k = s_k - s_{k-1},\dots,N_2 = s_2 - s_1, N_1 = s_1))}{\mathbb{P}(N_{k-1} = s_{k-1} - s_{k-2},\dots,N_2 = s_2 - s_1, N_1 = s_1)} \\
&= \mathbb{P}(N_k = s_k - s_{k-1} \mid N_{k-1} = s_{k-1} - s_{k-2},\dots,N_2 = s_2 - s_1, N_1 = s_1) \\
&= \mathrm{h}_{n-s_{k-1},m_k,m-s}(s_k - s_{k-1})
\end{aligned}$$

From this, with Theorem 1.14 we get the proposition. □

1.4 Application: The distribution of the multinomial range

Pfeifer [17] discussed inappropriate intuitive uses of the expectation μ in repeated trials each with probability of success μ. To clearify his argumentation, Pfeifer computed the probability density function of the Range

$$D_n := \max_{i=1}^d N_i - \min_{i=1}^d N_i$$

of a multinomially distributed random variable $(N_1,\dots,N_d) \sim \mathrm{M}_{n,p}$ for $n = 100, d = 6$ and $p = (1/d,\dots,1/d)$. To compute the probabilty density function of the Range D_n for $n = 1000$ and the same p, he made use of a simulating algorithm.

According to Corrado [4] this probability can be computed with the following formula, which uses rectangle probabilities

$$\begin{aligned}
\mathbb{P}(D_n \le k) = &\sum_{h=0}^{n-k} \mathbb{P}(N_1 \in \{h,\dots,h+k\},\dots,N_d \in \{h,\dots,h+k\}) \\
&- \sum_{h=0}^{n-k-1} \mathbb{P}(N_1 \in \{h+1,\dots,h+k\},\dots,N_d \in \{h+1,\dots,h+k\})
\end{aligned} \tag{1.3}$$

Table 1.1: The cumulative distribution function of the multinomial Range D_{1000}

k	$\mathbb{P}(D_{1000} \leq k)$	k	$\mathbb{P}(D_{1000} \leq k)$	k	$\mathbb{P}(D_{1000} \leq k)$	k	$\mathbb{P}(D_{1000} \leq k)$
1	$1.028242 \cdot 10^{-6}$	18	0.0863429	35	0.6253632	52	0.953737
2	$6.541602 \cdot 10^{-6}$	19	0.1059366	36	0.6576482	53	0.9605108
3	$3.882427 \cdot 10^{-5}$	20	0.1279544	37	0.6884817	54	0.9664172
4	0.0001275595	21	0.1523472	38	0.717738	55	0.9715444
5	0.0003384679	22	0.1790134	39	0.7453226	56	0.9759761
6	0.00075915	23	0.207805	40	0.7711718	57	0.9797904
7	0.001510181	24	0.2385296	41	0.7952502	58	0.9830595
8	0.002740486	25	0.2709564	42	0.8175492	59	0.9858498
9	0.00463207	26	0.304823	43	0.8380834	60	0.9882219
10	0.007381874	27	0.3398435	44	0.8568881	61	0.9902303
11	0.01121232	28	0.3757151	45	0.8740164	62	0.991924
12	0.01635051	29	0.4121272	46	0.8895357	63	0.9933469
13	0.02302565	30	0.4487688	47	0.9035247	64	0.9945377
14	0.03145717	31	0.4853358	48	0.916071	65	0.9955304
15	0.04184907	32	0.5215378	49	0.9272679	66	0.9963548
16	0.05437359	33	0.5571033	50	0.9372124	67	0.9970371
17	0.06917238	34	0.5917851	51	0.9460028	68	0.9975995

We use formula (1.3) in the R algorithm stated in Appendix A to compute the cumulative distribution function and with that the probability density function of the multinomial Range D_{1000}. The values $\mathbb{P}(D_{1000} \leq k)$ of the cumulative distribution function for $k \in \{1, \ldots, 68\}$ are listed in Table 1.1, while in Figure 1.1 a plot of the probability density function of D_{1000} is shown. This may be compared with Pfeifer's diagram on his p.6, obtained by simulation.

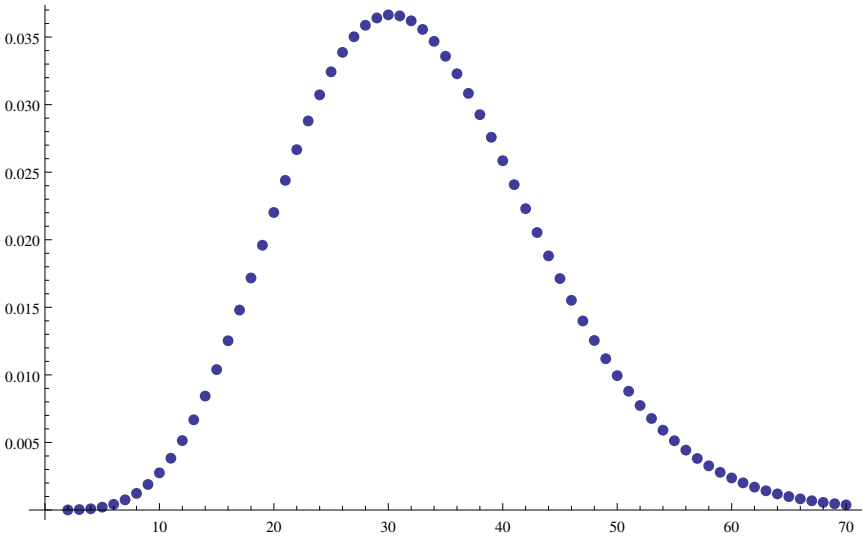

Figure 1.1: The probability density function of the multinomial Range D_{1000}

Chapter 2

Basics of approximative computations

Mathematical algorithms often are applied using computers that are not able to perform the operations $+, -, \cdot, /$ on $\mathbb{R}$ exactly, but approximatively. We are interested in the accuracy of results which we get when we use such a computer for applying the algorithms for Rectangle Probabilities that we derived in Chapter 1. As preparation for an analysis of their accuracy, in this chapter we state important basics of approximative computations.

2.1 Definitions of computer number systems and operations

In this section we define number systems and operations that we assume the considered computers use. We will work in a general setting, where computer numbers are elements of an ordered field, not nessesarily real numbers. As far as possible, we will not make any further assumptions about the structure of the number system or the distances between computer numbers but only assume that there exists a constant which bounds the relative error between the computed result and the exact result of an operation in the field. This will only be possible if the result lies in a subset of the ordered field. We will call such a subset the range of the computer number system. Similar approaches, which are the same in the important case of the number systems defined by the IEEE 754 Standard [3], can be found in [9] or [14].

In the entire rest of this work let $(K, +, \cdot, \leq)$ be an ordered field and $-\infty, \infty$ two objects with $-\infty, \infty \notin K$. We expand the order $\leq$ to the set $\overline{K} := K \cup \{-\infty, \infty\}$ by defining $-\infty \leq x$ and $x \leq \infty$ for every $x \in \overline{K}$. For $x, y \in \overline{K}$ we define $x < y :\Leftrightarrow y > x :\Leftrightarrow (x \leq y$ and $x \neq y)$ as well as $x \geq y :\Leftrightarrow y \leq x$. We define

$$\begin{aligned} -\infty + x := x + (-\infty) := -\infty \quad & \text{for } x \in K \cup \{-\infty\} \\ x + \infty := \infty + x := \infty \quad & \text{for } x \in K \cup \{\infty\} \end{aligned}$$

and

$$-\infty \cdot x := x \cdot (-\infty) \quad := \quad \begin{cases} -\infty & \text{for } x \in \overline{K} \text{ with } x > 0 \\ 0 & \text{for } x = 0 \\ \infty & \text{for } x \in \overline{K} \text{ mit } x < 0 \end{cases}$$

$$\infty \cdot x := x \cdot \infty \quad := \quad \begin{cases} \infty & \text{for } x \in \overline{K} \text{ with } x > 0 \\ 0 & \text{for } x = 0 \\ -\infty & \text{for } x \in \overline{K} \text{ with } x < 0 \end{cases}$$

We further define $-y := \infty$ for $y = -\infty$, $y^{-1} := 0$ for $y \in \{-\infty, \infty\}$ and $y^{-1} := \infty$ for $y = 0$ and with that $x/y := \frac{x}{y} := x \cdot (y^{-1})$ for $(x, y) \in \overline{K}^2$ and $x - y := x + (-y)$ for $(x, y) \in \overline{K}^2 \setminus \{(-\infty, -\infty), (\infty, \infty)\}$. We further define the sets $D_+ := \overline{K}^2 \setminus \{(-\infty, \infty), (\infty, -\infty)\}$, $D_- := \overline{K}^2 \setminus \{(-\infty, -\infty), (\infty, \infty)\}$ and $D_* := K^2$ for $* \in \{\cdot, /\}$. With these sets we just defined functions $* : D_* \to \overline{K}$ for $* \in \{+, -, \cdot, /\}$. For $x \in \overline{K}$ let the **absolute value** $|x| := x$ if $x > 0$ and $|x| := -x$ if $x \leq 0$. The set $\mathbb{N} = \mathbb{N}_K = \{1, 2, 3, \ldots\}$ as well as intervalls in K we define in usual way. For $x, \tilde{x} \in K$ we define the **relative error** $\mathrm{e_{rel}}(x, \tilde{x}) := |x - \tilde{x}|/|x|$. For $A \subseteq \overline{K}$ we set $-A := \{-x : x \in A\}$ and $\pm A := (-A) \cup A$. Let NaN denote a set which is not element of the set $\overline{K}$. The symbol NaN stands as an abbreviation for "Not a Number" and will be used as a computer number which a computer returns as exceptional result.

Now we define roundings into a finite subset of K.

Definition 2.1. For a finite subset $M \subseteq K$ we define $\overline{M} := M \cup \{-\infty, \infty\}$ and the functions $\mathrm{rd}_M, \mathrm{ru}_M : \overline{K} \to \overline{M}$ by

$$\mathrm{rd}_M(x) := \max\{z \in \overline{M} : z \leq x\}$$

$$\mathrm{ru}_M(x) := \min\{z \in \overline{M} : z \geq x\}$$

for $x \in \overline{K}$. Every function $r : \overline{K} \to \overline{M}$ we call **rounding** into M. The function rd_M we call **lower rounding** and the function ru_M we call **upper rounding** into M. A rounding $r : \overline{K} \to \overline{M}$ we call **regular**, if $\mathrm{rd}_M \leq r \leq \mathrm{ru}_M$, and **monotonic**, if $r(x) \leq r(\tilde{x})$ for every $x, \tilde{x} \in \overline{K}$ with $x \leq \tilde{x}$.

In the rest of this section let always $M \subseteq K$ be a finite subset of K and $\mathrm{rd} := \mathrm{rd}_M, \mathrm{ru} := \mathrm{ru}_M$.

We now define computer operations for approximative computations with results rounded downwards or rounded upwards, respectively.

Definition 2.2. Let $C := \overline{K} \cup \{\mathrm{NaN}\}$. For $* \in \{+, -, \cdot, /\}$ we define the functions $\underline{\circledast}_M, \overline{\circledast}_M : C^2 \to C$ by

$$x \underline{\circledast}_M y := x \overline{\circledast}_M y := \mathrm{NaN} \text{ for } (x, y) \in C^2 \setminus D_*$$

$$x\underline{\circledast}_M y := \mathrm{rd}(x * y), x\overline{\circledast}_M y := \mathrm{ru}(x * y) \text{ for } (x, y) \in D_*$$

with the following exceptions

$$x\underline{\odot}_M y := x\overline{\odot}_M y := \mathrm{NaN} \text{ if } x = 0, y \in \{-\infty, \infty\} \text{ or } y = 0, x \in \{-\infty, \infty\}$$

$$x\underline{\oslash}_M y := x\overline{\oslash}_M y := \mathrm{NaN} \text{ if } x = y = 0 \text{ or } x, y \in \{-\infty, \infty\}$$

We note that from the last definition we for example get $0\underline{\odot}_M\infty = 0\overline{\odot}_M\infty = \mathrm{NaN}$ while $0 \cdot \infty = 0$. These and further exceptions in the last definition are consistent with the definition of the computer operations according to the IEEE Standard 754.

In the rest of this section, we study roundings and computer operations. We declare that computer operations $\oplus, \ominus, \odot, \oslash$ always have higher priority than each of the operations $+, -, \cdot, /$, so for example the expression $a \cdot b - c$ means $(a \cdot b) - c$ and the expression $a \cdot b \ominus c$ means $a \cdot (b \ominus c)$.

Lemma 2.3. *For $x \in \overline{K}$ we have the properties $\mathrm{rd}(x) \leq x \leq \mathrm{ru}(x)$ and*

$$\forall z \in \overline{M} : x \geq z \Rightarrow \mathrm{rd}(x) \geq z$$

$$\forall z \in \overline{M} : x \leq z \Rightarrow \mathrm{ru}(x) \leq z$$

and if $\mathrm{rd}(x) \neq \mathrm{ru}(x)$ we have

$$\mathrm{ru}(x) = \min\{z \in \overline{M} : z > \mathrm{rd}(x)\}$$

Proof. We have $\mathrm{rd}(x) = \max\{w \in \overline{M} : w \leq x\} \leq x$. For $z \in \overline{M}$ with $z \leq x$ we have $z \leq \max\{w \in \overline{M} : w \leq x\} = \mathrm{rd}(x)$. The inequality $z \geq \mathrm{ru}(x)$ for every $z \in \overline{M}$ with $z \geq x$ follows from $\mathrm{ru}(x) = -\mathrm{rd}_{-M}(-x)$. We get $\mathrm{rd}(x) \leq x \leq \mathrm{ru}(x)$. For every $z \in \overline{M}$ with $z > \mathrm{rd}(x)$ we have $z > x$ and therefore $z \geq \mathrm{ru}(x)$. If $\mathrm{rd}(x) \neq \mathrm{ru}(x)$ then we have $\mathrm{rd}(x) < \mathrm{ru}(x)$ and therefore $\mathrm{ru}(x) = \min\{z \in \overline{M} : z > \mathrm{rd}(x)\}$. □

From the last Lemma we further get the following corollary.

Corollary 2.4. *For every regular rounding $r : \overline{K} \to \overline{M}$ and $x \in \overline{K}, z \in \overline{M}$ we have the implications*

$$x \geq z \Rightarrow r(x) \geq z$$

$$x \leq z \Rightarrow r(x) \leq z$$

Proof. From Lemma 2.3 we have

$$x \geq z \Rightarrow r(x) \geq \mathrm{rd}(x) \geq z$$

$$x \leq z \Rightarrow r(x) \leq \mathrm{ru}(x) \leq z$$

□

Corollary 2.5. *The lower rounding* rd *and the upper rounding* ru *are regular and monotonic.*

Proof. The regularity of rd, ru follows from $\mathrm{rd} \leq \mathrm{ru}$. Let $x, \tilde{x} \in \overline{K}$ with $x \leq \tilde{x}$. Then $\mathrm{ru}(x) = \min\{z \in \overline{M} : z \geq x\} \leq \mathrm{ru}(\tilde{x})$ and $\mathrm{rd}(x) = \max\{z \in \overline{M} : z \leq x\} \leq \mathrm{rd}(\tilde{x})$. Therefore rd, ru are monotonic □

Definition 2.6. A rounding $r : \overline{K} \to \overline{M}$ we call **close**, if there exist $A, B \in K$ with $]A, B[\supseteq M$ and

$$|r(x) - x| = \min\{|\mathrm{ru}(x) - x|, |\mathrm{rd}(x) - x|\} \tag{2.1}$$

for every $x \in]A, B[$ and $r(x) = -\infty$ if $x \in [-\infty, A]$ and $r(x) = \infty$ if $x \in [B, \infty]$.

Lemma 2.7. *Every close rounding is regular and monotonic*

Proof. Let $r : \overline{K} \to \overline{M}$ a close rounding and $A, B \in K$ with $]A, B[\supseteq M$ and (2.1) for every $x \in]A, B[$ and $r(x) = -\infty$ if $x \in [-\infty, A]$ and $r(x) = \infty$ if $x \in [B, \infty]$.

Proof of r being regular: If $x \in [-\infty, A]$ we have $r(x) = -\infty = \mathrm{rd}(x)$, if $x \in [B, \infty]$ we have $r(x) = \infty = \mathrm{ru}(x)$. Let $x \in]A, B[$. If $r(x) \geq x$, with Lemma 2.3 we get $r(x) \geq \mathrm{ru}(x) \geq x$ and therefore $|r(x) - x| \geq |\mathrm{ru}(x) - x|$. Because of (2.1) we also have $|r(x) - x| \leq |\mathrm{ru}(x) - x|$ and therefore $|r(x) - x| = |\mathrm{ru}(x) - x|$ which is equivalent to $r(x) = \mathrm{ru}(x)$. If $r(x) \leq x$, with Lemma 2.3 we get $r(x) \leq \mathrm{rd}(x) \leq x$ and therefore $|r(x) - x| \geq |\mathrm{rd}(x) - x|$. Because of (2.1) we also have $|r(x) - x| \leq |\mathrm{rd}(x) - x|$ and therefore $|r(x) - x| = |\mathrm{rd}(x) - x|$ and therefore $r(x) = \mathrm{rd}(x)$.

Proof of r being monotonic: Let $x, \tilde{x} \in \overline{K}$ with $x < \tilde{x}$. We prove $r(x) \leq r(\tilde{x})$. Because r is regular we have $r(x) \in \{\mathrm{rd}(x), \mathrm{ru}(x)\}$ and $r(\tilde{x}) \in \{\mathrm{rd}(\tilde{x}), \mathrm{ru}(\tilde{x})\}$. In case of $r(x) = \mathrm{rd}(x)$ we have $r(x) = \mathrm{rd}(x) \leq \mathrm{rd}(\tilde{x}) \leq r(\tilde{x})$ while in case of $r(x) = \mathrm{ru}(x)$ and $r(\tilde{x}) = \mathrm{ru}(\tilde{x})$ we have $r(x) = \mathrm{ru}(x) \leq \mathrm{ru}(\tilde{x}) = r(\tilde{x})$. Now let $r(x) = \mathrm{ru}(x)$ and $r(\tilde{x}) = \mathrm{rd}(\tilde{x})$. In this case we have $x, \tilde{x} \in]A, B[$ because otherwise it would be $r(x) = -\infty$ or $r(\tilde{x}) = \infty$ and therefore $r(x) \neq \mathrm{ru}(x)$ or $r(\tilde{x}) \neq \mathrm{rd}(\tilde{x})$. From $r(\tilde{x}) = \mathrm{rd}(\tilde{x})$ with (2.1) we get $|\mathrm{rd}(\tilde{x}) - \tilde{x}| \leq |\mathrm{ru}(\tilde{x}) - \tilde{x}|$. Assumed that $\mathrm{ru}(x) > \mathrm{rd}(\tilde{x})$ from Lemma 2.3 we would get $\mathrm{ru}(x) \geq \mathrm{ru}(\tilde{x})$ and $\mathrm{rd}(\tilde{x}) \leq \mathrm{rd}(x)$, and with $\mathrm{ru}(x) \leq \mathrm{ru}(\tilde{x})$ and $\mathrm{rd}(x) \leq \mathrm{rd}(\tilde{x})$ we would get $\mathrm{ru}(x) = \mathrm{ru}(\tilde{x})$ and $\mathrm{rd}(x) = \mathrm{rd}(\tilde{x})$. Therefore with $|\mathrm{rd}(x) - \tilde{x}| = |\mathrm{rd}(\tilde{x}) - \tilde{x}| \leq |\mathrm{ru}(\tilde{x}) - \tilde{x}| = |\mathrm{ru}(x) - \tilde{x}|$ and $x < \tilde{x}$ we would get $|\mathrm{rd}(x) - x| < |\mathrm{ru}(x) - x|$ which contradicts $r(x) = \mathrm{ru}(x)$. Thus we have $\mathrm{ru}(x) \leq \mathrm{rd}(\tilde{x})$ and with that the proposition $r(x) = \mathrm{ru}(x) \leq \mathrm{rd}(\tilde{x}) = r(\tilde{x})$. □

Theorem 2.8. *Let* $r : \overline{K} \to \overline{M}$ *a monotonic rounding and for* $* \in \{+, -, \cdot, /\}$ *let* $\circledast : D_* \to \overline{M}$ *defined by*

$$x \circledast y := r(x * y)$$

for $(x, y) \in D_*$.

Then for $(x, y), (\tilde{x}, \tilde{y}) \in D_*$ *we have the implications*

(a) Monoticity of $\oplus$

$$x \leq \tilde{x} \text{ and } y \leq \tilde{y} \Rightarrow x \oplus y \leq \tilde{x} \oplus \tilde{y} \tag{2.2}$$

(b) Monoticity of $\ominus$

$$x \leq \tilde{x} \text{ and } y \geq \tilde{y} \Rightarrow x \ominus y \leq \tilde{x} \ominus \tilde{y} \tag{2.3}$$

(c) Monoticites of $\odot$

$$0 \leq x \leq \tilde{x} \text{ and } 0 \leq y \leq \tilde{y} \Rightarrow x \odot y \leq \tilde{x} \odot \tilde{y} \tag{2.4}$$

$$0 \geq x \geq \tilde{x} \text{ and } 0 \leq y \leq \tilde{y} \Rightarrow x \odot y \geq \tilde{x} \odot \tilde{y} \tag{2.5}$$

$$0 \geq x \geq \tilde{x} \text{ and } 0 \geq y \geq \tilde{y} \Rightarrow x \odot y \leq \tilde{x} \odot \tilde{y} \tag{2.6}$$

$$0 \leq x \leq \tilde{x} \text{ and } 0 \geq y \geq \tilde{y} \Rightarrow x \odot y \geq \tilde{x} \odot \tilde{y} \tag{2.7}$$

(d) Monoticites of $\oslash$

$$0 \leq x \leq \tilde{x} \text{ and } 0 < \tilde{y} \leq y \Rightarrow x \oslash y \leq \tilde{x} \oslash \tilde{y} \tag{2.8}$$

$$0 \geq x \geq \tilde{x} \text{ and } 0 < \tilde{y} \leq y \Rightarrow x \oslash y \geq \tilde{x} \oslash \tilde{y} \tag{2.9}$$

$$0 \geq x \geq \tilde{x} \text{ and } 0 > \tilde{y} \geq y \Rightarrow x \oslash y \leq \tilde{x} \oslash \tilde{y} \tag{2.10}$$

$$0 \leq x \leq \tilde{x} \text{ and } 0 > \tilde{y} \geq y \Rightarrow x \oslash y \geq \tilde{x} \oslash \tilde{y} \tag{2.11}$$

Proof. In each case, under the stated condition we get $x * y \leq \tilde{x} * \tilde{y}$ or $\tilde{x} * \tilde{y} \leq x * y$ and with the monoticity of r we get the propositions. □

Lemma 2.9. *Let* $r : \overline{K} \to \overline{M}$ *a regular rounding and for* $* \in \{+, -, \cdot, /\}$ *let* $\circledast : D_* \to \overline{M}$ *defined by*

$$x \circledast y := r(x * y)$$

for $(x, y) \in D_*$.
Then for $(x, y) \in D_*$ *and* $z \in \overline{M}$ *we get the following implications.*

$$x * y \leq z \Rightarrow x \circledast y \leq z \tag{2.12}$$

$$x * y \geq z \Rightarrow x \circledast y \geq z \tag{2.13}$$

$$x * y = z \Rightarrow x \circledast y = z \tag{2.14}$$

Proof. The first two implications follow from Corollary 2.4. If $x * y = z$ then $x * y \leq z$ and $x * y \geq z$ and therefore we get $x \circledast y = z$. □

In the IEEE-Standard 754 the computer operations $\circledast$ for $* \in \{+, -, \cdot, /\}$ are not consistent with our definition of the operations $+, -, \cdot, /$ in K but there are some exceptions. These exceptions are

$$x \odot y = \text{NaN if } x = 0, y \in \{-\infty, \infty\} \text{ or } x \in \{-\infty, \infty\}, y = 0$$

while our definition of the operation $\cdot$ in K yields

$$x \cdot y = 0 \text{ if } x = 0, y \in \{-\infty, \infty\} \text{ or } x \in \{-\infty, \infty\}, y = 0$$

Further exceptions are

$$x \oslash y = \text{NaN if } x = y = 0 \text{ or } x, y \in \{-\infty, \infty\}$$

while our definition of the operation $/$ in K yields

$$x/y = 0 \text{ if } x = y = 0 \text{ or } x, y \in \{-\infty, \infty\}$$

In the following two theorems we examine if the implications (2.2) - (2.14) still remain true when we define the operations $\circledast$ with the exceptions from the IEEE-Standard 754.

Corollary 2.10. *Let $r : \overline{K} \to \overline{M}$ a monotonic rounding and $C := \overline{K} \cup \text{NaN}$. For $* \in \{+, -, \cdot, /\}$ let the functions $\circledast_M : C^2 \to C$ be defined in the following way:*

$$x \circledast_M y := \text{NaN } \textit{for } (x, y) \in C^2 \setminus D_*$$

$$x \circledast_M y := r(x * y) \textit{ for } (x, y) \in D_*$$

with the following exceptions

$$x \odot_M y := \text{NaN } \textit{if } x = 0, y \in \{-\infty, \infty\} \textit{ or } x \in \{-\infty, \infty\}, y = 0$$

$$x \oslash_M y := \text{NaN } \textit{if } x = y = 0 \textit{ or } x, y \in \{-\infty, \infty\}$$

Then for $x, y, \tilde{x}, \tilde{y} \in K$ and $* \in \{+, -, \cdot, /\}$ we get the implications (2.2) - (2.11) *with $\circledast = \circledast_M$.*

Proof. Let $* \in \{+, -, \cdot, /\}$ and $\tilde{\circledast} : D_* \to \overline{M}$ without exceptions defined by

$$x \tilde{\circledast} y := r(x * y) \text{ for } (x, y) \in D_*$$

From Theorem 2.8 we get the implications (2.2) - (2.11) for $(x, y), (\tilde{x}, \tilde{y}) \in D_*$ and $\circledast = \tilde{\circledast}$. For $* \in \{+, -, \cdot\}$ and $x, y, \tilde{x}, \tilde{y}$ or for $* = /$ and $x, \tilde{x} \in K, y, \tilde{y} \in K \setminus \{0\}$ we have $(x, y), (\tilde{x}, \tilde{y}) \in D_*$ and $x \circledast_M y = x \tilde{\circledast} y$ and $\tilde{x} \circledast_M \tilde{y} = \tilde{x} \tilde{\circledast} \tilde{y}$. Therefore we get the proposed implications. □

Corollary 2.11. *Let $r : \overline{K} \to \overline{M}$ a regular rounding and $C := \overline{K} \cup \mathrm{NaN}$. For $* \in \{+, -, \cdot, /\}$ let the functions $\circledast_M : C^2 \to C$ be defined in the following way:*

$$x \circledast_M y := \mathrm{NaN} \text{ for } (x, y) \in C^2 \setminus D_*$$

$$x \circledast_M y := r(x * y) \text{ for } (x, y) \in D_*$$

with the following exceptions

$$x \odot_M y := \mathrm{NaN} \text{ if } x = 0, y \in \{-\infty, \infty\} \text{ or } x \in \{-\infty, \infty\}, y = 0$$

$$x \oslash_M y := \mathrm{NaN} \text{ if } x = y = 0 \text{ or } x, y \in \{-\infty, \infty\}$$

Then for $* \in \{+, -, \cdot, /\}$ and $x, y \in K, z \in \overline{M}$ with $y \neq 0$ in case of $* = /$ we get the implications (2.12) - (2.14) *with $\circledast = \circledast_M$.*

Proof. Let $* \in \{+, -, \cdot, /\}$ and $\tilde{\circledast} : D_* \to \overline{M}$ without exceptions defined by

$$x \tilde{\circledast} y := r(x * y) \text{ for } (x, y) \in D_*$$

From Lemma 2.9 we get the implications (2.2) - (2.11) for $(x, y) \in D_*, z \in \overline{M}$ and $\circledast = \tilde{\circledast}$. For $x, y \in K, z \in \overline{M}$ with $y \neq 0$ in case of $* = /$ we have $x \circledast_M y = r(x * y) = x \tilde{\circledast} y$ and therefore the implications (2.12) - (2.14) with $\circledast = \circledast_M$. □

We want to derive accuracy bounds for rounded computations. For this purpose, we define the following approximating property of roundings.

Definition 2.12. Let $R \subseteq K \setminus \{0\}$ and $u \in [0, \infty[$. A rounding $r : \overline{K} \to \overline{M}$ is called an (R, u)**-approximator**, if

$$r(x) \in K \text{ and } \mathrm{e}_{\mathrm{rel}}(x, r(x)) \leq u \tag{2.15}$$

for every $x \in R$.

Now we derive accuracy bounds for rounded computations. First we only consider rounded results in a subset F of positive elements of M.

Lemma 2.13. *Let $F \subseteq M \cap]0, \infty[$ with $\#(F) \geq 2$. For $z \in F$ with $z < \max F$ we define $\mathrm{succ}(z) := \min\{y \in F : y > z\}$. Let*

$$u := \frac{1}{2} \max \{\mathrm{e}_{\mathrm{rel}}(z, \mathrm{succ}(z)) : z \in F \setminus \{\max F\}\}$$

and

$$R := \left[\min F, \max F + \frac{1}{2}(\max F - \max(F \setminus \{\max F\}))\right[$$

Let $r : \overline{K} \to \overline{M}$ a rounding with (2.1) *for $x \in R$. Then r is an (R, u)-approximator.*

Proof. Let $x \in R$. The condition $r(x) \in K$ obviously is true. If $x \leq \max F$ then $\mathrm{ru}(x) \leq \max F$ and

$$|r(x) - x| = \min\{|\mathrm{ru}(x) - x|, |\mathrm{rd}(x) - x|\} \leq \frac{1}{2}|\mathrm{rd}(x) - \mathrm{ru}(x)|$$

and therefore $\mathrm{e_{rel}}(x, r(x)) \leq \mathrm{e_{rel}}(\mathrm{rd}(x), \mathrm{ru}(x))/2 \leq u$. If $x > \max F$ then $\mathrm{rd}(x) \geq \max F$ and we have

$$x - \mathrm{rd}(x)) < \frac{1}{2}(\max F - \max(F \setminus \{\max F\})$$

and therefore $\mathrm{e_{rel}}(x, r(x)) \leq \frac{1}{2}\mathrm{e_{rel}}(\max(F \setminus \{\max F\}), \max F) \leq u$. □

The following Theorem 2.14 generalizes the result of Lemma 2.13 by allowing negative elements in the subset $F \subseteq M$.

Theorem 2.14. *Let $F \subseteq M$ with $\#(F \cap]-\infty, 0[) \geq 2$ and $\#(F \cap]0, \infty[) \geq 2$. For $z \in F$ with $z > \min F$ we define $\mathrm{prec}(z) := \max\{y \in F : y < z\}$ and for $z \in F$ with $z < \max F$ we define $\mathrm{succ}(z) := \min\{y \in F : y > z\}$. Let*

$$v := 1/2 \max\{\mathrm{e_{rel}}(z, \mathrm{prec}(z)) : z \in F \cap]\min F, 0[\}$$

$$w := 1/2 \max\{\mathrm{e_{rel}}(z, \mathrm{succ}(z)) : x \in F \cap]0, \max F[\}$$

$$R_1 :=]\min F - 1/2(\min(F \setminus \{\min F\}) - \min F), \max(F \cap]-\infty, 0[)]$$

$$R_2 := [\min(F \cap]0, \infty[), \max F + 1/2(\max F - \max(F \setminus \{\max F\}))[$$

Let $u := \max\{v, w\}$ and $R := R_1 \cup R_2$. Let $r : \overline{K} \to \overline{M}$ a rounding with (2.1) *for $x \in R$. Then r is an (R, u)-approximator.*

Proof. From Lemma 2.13 we directly get

$$\mathrm{e_{rel}}(x, r(x)) \leq w \text{ for } x \in R_2$$

If we apply Lemma 2.13 to the functions $\mathrm{ru}_N, \mathrm{rd}_N$ with $N := -M$ instead of M, the set $G := -(F \cap]-\infty, 0[)$ instead of F, and the function $q : -R_1 \to \overline{N}$ defined by $q(x) := -r(-x)$ for $x \in -R_1$, we get

$$\mathrm{e_{rel}}(x, r(x)) \leq v \text{ for } x \in M \text{ with } x \in R_1$$

because of $\mathrm{ru}_N(-x) = -\mathrm{rd}(x), \mathrm{rd}_N(-x) = -\mathrm{ru}(x)$ and $\mathrm{e_{rel}}(-x, q(-x)) = \mathrm{e_{rel}}(x, r(x))$ for $x \in R_1$. Therefore the proposition is proven. □

We remark that in case of F being symmetrical, that means $F = \pm F$, in Theorem 2.14 we get the easier expressions $\max\{v, w\} = v$ and $R_1 \cup R_2 = \pm R_1$.

Corollary 2.15. *Let $R \subseteq K \setminus \{0\}, u \in [0, 1[$ and $r : \overline{K} \to \overline{M}$ an (R, u)-approximator. Let $C := \overline{K} \cup \mathrm{NaN}$. For $* \in \{+, -, \cdot, /\}$ let the functions $\circledast_M : C^2 \to C$ be defined in the following way*

$$x \circledast_M y := \mathrm{NaN} \text{ for } (x, y) \in C^2 \setminus D_*$$

$$x \circledast_M y := r(x * y) \text{ for } (x, y) \in D_*$$

with the following exceptions

$$x \odot_M y := \mathrm{NaN} \text{ if } x = 0, y \in \{-\infty, \infty\} \text{ or } x \in \{-\infty, \infty\}, y = 0$$

$$x \oslash_M y := \mathrm{NaN} \text{ if } x = y = 0 \text{ or } x, y \in \{-\infty, \infty\}$$

Then for $* \in \{+, -, \cdot, /\}$ and for $x, y \in K$ with $x * y \in R$ we have

$$x \circledast_M y \in K \text{ and } \mathrm{e_{rel}}(x * y, x \circledast_M y) \leq u$$

Proof. Let $* \in \{+, -, \cdot, /\}$ and $x, y \in K$ with $x * y \in R$. Because of $x, y \in K$ we have $(x, y) \in D_*$. Because of $0 \notin R$ and $x * y \in R$ we have $x * y \neq 0$. Therefore we have got none of the exceptional cases in the definition of $\circledast_M$, but $x \circledast_M y = r(x * y)$. Because of $x * y \in R$ we have $r(x * y) \in K$ and $\mathrm{e_{rel}}(x * y, r(x * y)) \leq u$. Therefore we get the proposition. □

2.2 The computer number systems $C_{s,t}$

We now define computer number systems $C_{s,t}$ with $s, t \in \mathbb{N}$. For $s = 11, t = 52$ we get the number system IEEE-Double and for $s = 8, t = 23$ we get the number system IEEE-Single which are described in the IEEE Standard 754 [3]. Particularly the first of these two examples is important because of frequent applications in practice.

Definition 2.16. Let $s, t \in \mathbb{N}$ and $C_{s,t} := \pm F_{s,t} \cup \pm G_{s,t} \cup \{-\infty, \infty, \mathrm{NaN}\}$ with

$$F_{s,t} := \{(1 + d\varepsilon_t)2^{e-e_0} : d \in \{0, \ldots, 2^t - 1\}, e \in \{1, \ldots, 2^s - 2\}\}$$

with $\varepsilon_t := 2^{-t}$ and $e_0 := 2^{s-1} - 1$, and

$$G_{s,t} = \left\{d\varepsilon_t 2^{e_{\min}} : d \in \{0, \ldots, 2^t - 1\}\right\}$$

with $e_{\min} := 1 - e_0 = -2^{s-1} + 2$. Let

$$R_{s,t} := \pm[\min F_{s,t}, \max F_{s,t} + 2^{2^{s-1}-t-2}[$$

and $\mathrm{u}_t := 2^{-(t+1)}$. We define $\mathrm{IEEESingle} := C_{8,23}, \mathrm{IEEEDouble} := C_{11,52}$ and $\mathrm{IEEEMinExtended} := C_{15,63}$.

For $s,t \in \mathbb{N}$ we define $\mathrm{rd}_{s,t} := \mathrm{rd}_{\pm F_{s,t} \cup \pm G_{s,t}}$ and $\mathrm{ru}_{s,t} := \mathrm{ru}_{\pm F_{s,t} \cup \pm G_{s,t}}$. For $* \in \{+,-,\cdot,/\}$ we define

$$\overline{\circledast}_{s,t} := \overline{\circledast}_{\pm F_{s,t} \cup \pm G_{s,t}}, \underline{\circledast}_{s,t} := \underline{\circledast}_{\pm F_{s,t} \cup \pm G_{s,t}}$$

We further define

$$A_{s,t} := \max F_{s,t} + \frac{1}{2}(\max F_{s,t} - \max(F_{s,t} \setminus \{\max F_{s,t}\}))$$

In this section we always assume $r_{s,t} : \overline{K} \to C_{s,t} \cap \overline{K}$ is a rounding with

$$|x - r_{s,t}(x)| = \min\{|x - \mathrm{rd}_{s,t}(x)|, |x - \mathrm{ru}_{s,t}(x)|\}$$

for $x \in]-A_{s,t}, A_{s,t}[$ and with $r_{s,t}(x) = -\infty$ for $x \in [-\infty, -A_{s,t}]$ and $r_{s,t}(x) = \infty$ for $x \in [A_{s,t}, \infty]$. Further let $C := \overline{K} \cup \{\mathrm{NaN}\}$ and for $* \in \{+,-,\cdot,/\}$ the functions $\circledast_{s,t} : C^2 \to C_{s,t}$ defined as

$$x \circledast_{s,t} y := \mathrm{NaN} \text{ for } (x,y) \in C^2 \setminus D_*$$

$$x \circledast_{s,t} y := r_{s,t}(x * y) \text{ for } (x,y) \in D_*$$

with the following exceptions

$$x \odot_{s,t} y := \mathrm{NaN} \text{ if } x = 0, y \in \{-\infty, \infty\} \text{ or } x \in \{-\infty, \infty\}, y = 0$$

$$x \oslash_{s,t} y := \mathrm{NaN} \text{ if } x = y = 0 \text{ or } x, y \in \{-\infty, \infty\}$$

We list the smallest examples of sets of the form $F_{s,t}$.

Example 2.17. For every $t \in \mathbb{N}$ we have $F_{1,t} = \emptyset$. We have

$$F_{2,1} = \{1, 1.5, 2, 3\}, R_{2,1} = \pm[1, 3.5[, u_1 = 1/4$$

As $F_{q,r} \subseteq F_{s,t}$ if $q,r,s,t \in \mathbb{N}$ with $q \leq s$ and $r \leq t$, this means that $1, 1.5, 2, 3 \in F_{s,t}$ for $s,t \in \mathbb{N}$ with $s \geq 2$. For example we have

$$F_{2,2} = \{1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5\}, R_{2,2} = \pm[1, 3.75[, u_2 = 1/8$$

$$F_{3,1} = \{0.25, 0.375, 0.5, 0.75, 1, 1.5, 2, 3, 4, 6, 8, 12\}, R_{3,1} = \pm[0.25, 14[$$

Lemma 2.18. *Let $s,t \in \mathbb{N}$ with $s \geq 2$. Then*

$$\min F_{s,t} = 2^{2-2^{s-1}}, \max F_{s,t} = (2 - 2^{-t})2^{2^{s-1}-1}$$

Particularly we have

$$[2^{2-2^{s-1}}, 2^{2^{s-1}-1}] \subseteq R_{s,t}$$

For $k \in \mathbb{Z}$ we have

$$2^k \in F_{s,t} \Leftrightarrow 2 - 2^{s-1} \leq k \leq 2^{s-1} - 1$$

$$2^k \in C_{s,t} \Leftrightarrow 2 - 2^{s-1} - t \leq k \leq 2^{s-1} - 1$$

Proof. With $d = 0$ and $e = 1$ we have $\min F_{s,t} = (1 + d\varepsilon_t)2^{e-(2^{s-1}-1)} = 2^{2-2^{s-1}}$ and with $d = 2^t - 1$ and $e = 2^s - 2$ we get $\max F_{s,t} = (1 + d\varepsilon_t)2^{e-(2^{s-1}-1)} = (2 - 2^{-t})2^{2^{s-1}-1}$. Let $k \in \{2 - 2^{s-1} - t, \ldots, 1 - 2^{s-1}\}$. Then with $d := 2^{2^{s-1}-2+t+k} \in \{0, \ldots, 2^{t-1}\}$ we have $2^k = d\varepsilon_t 2^{2-2^{s-1}}$. Hence $2^k \in G_{s,t}$. □

We now compute the values which in Theorem 2.14 we called u and R, in case of $F = F_{s,t}$, and get that $r_{s,t}$ is a $(R_{s,t}, u_t)$-approximator.

Lemma 2.19. *Let $s, t \in \mathbb{N}$. The rounding $r_{s,t}$ is a $(R_{s,t}, u_t)$-approximator. Therefore for $* \in \{+, -, \cdot, /\}$ and for $x, y \in K$ with $x * y \in R_{s,t}$ we have*

$$x \circledast_{s,t} y \in K \text{ and } \mathrm{e}_{\mathrm{rel}}(x * y, x \circledast_{s,t} y) \leq u_{s,t}$$

Proof. Let $F := F_{s,t}$. For $z \in F$ with $z < \max F$ we define $z' := \min\{y \in F : y > z\}$. Let $z \in F \setminus \{\max F\}$ and $e \in \{-2^{s-1}+2, \ldots, 2^{s-1}-1\}$ and $d \in \{0, \ldots, 2^t - 1\}$ with $z = (1+d\varepsilon_t)2^e$. Then $z' = (1 + (d + 1)\varepsilon_t)2^e$ and

$$\mathrm{e}_{\mathrm{rel}}(z, z') = \frac{\varepsilon_t 2^e}{z} = \frac{\varepsilon_t}{1 + d\varepsilon_t} \leq \varepsilon_t = 2^{-t}$$

Therefore

$$1/2 \max\{\mathrm{e}_{\mathrm{rel}}(z, z') : z \in F \setminus \{\max F\}\} \leq 2^{-t-1} = u_t$$

From Lemma 2.18 we have $\max F_{s,t} = (2-2^{-t})2^{2^{s-1}-1}$. We further have $\max(F \setminus \{\max F\}) = (2 - 2^{-t+1})2^{2^{s-1}-1}$. From that we get

$$1/2(\max F - \max(F \setminus \{\max F\})) = 2^{2^{s-1}-t-2}$$

With Theorem 2.14 we get that the rounding $r_{s,t}$ is a $(R_{s,t}, u_t)$-approximator. With that, from Corollary 2.15 we get that for $* \in \{+, -, \cdot, /\}$ and for $x, y \in K$ with $x * y \in R_{s,t}$ we have

$$x \circledast_{s,t} y \in K \text{ and } \mathrm{e}_{\mathrm{rel}}(x * y, x \circledast_{s,t} y) \leq u_{s,t}$$

□

We remark that Lemma 2.19 states that the precision u_t of the number system $C_{s,t}$ is determined by the parameter t and, if t is small compared to 2^s, the parameter s roughly determines how large the range $R_{s,t}$ of the number system $C_{s,t}$ is.

Lemma 2.20. *Let $t < 2^{s-1} - 1$. Then we have $\max\{n \in \mathbb{N} : \{1, \ldots, n\} \subseteq C_{s,t}\} = 2^{t+1}$.*

Proof. Let $k \in \{0, \ldots, t\}$ and $m \in \{0, \ldots, 2^k - 1\}$. We show that $2^k + m \in C_{s,t}$. Let $d := m2^{t-k}$ and $e := e_0 + k$. Then $d \in \{0, \ldots, 2^t - 1\}, e \in \{e_0, \ldots, 2e_0 - 1\}$ and $(1 + d\varepsilon_t)2^{e-e_0} = 2^{e-e_0} + m2^{t-k}\varepsilon_t 2^{e-e_0} = 2^k + m$. Furthermore we have $2^{t+1} = (1+d\varepsilon_t)2^{e-e_0}$ with $d = 0, e = e_0+t+1 \in \{e_0, \ldots, 2e_0\}$. For $d = 1, e = e_0 + t + 1$ we have $(1 + d\varepsilon_t)2^{e-e_0} = (1 + 2^{-t})2^{t+1} = 2^{t+1} + 2$. Hence $2^{t+1} + 1 \notin C_{s,t}$. □

Lemma 2.21. *Let $s,t \in \mathbb{N}$ and $x \in C_{s,t} \cap K$. If $|x| < 2^{2^{s-1}-1}$, then $-2x, 2x \in C_{s,t}$. If $|x| \geq 2^{3-2^{s-1}}$ then we have $-x/2, x/2 \in C_{s,t}$.*

Proof. If $2^{2-2^{s-1}} \leq |x| < 2^{2^{s-1}-1}$, then $x \in \pm F_{s,t}$ and there exist $d \in \{0,\dots,2^t-1\}, e \in \{2-2^{s-1},\dots,2^{s-1}-2\}$ with $|x| = (1+d\varepsilon_t)2^e$. Therefore $|2x| = (1+d\varepsilon_t)2^{e+1}$, with $e+1 \leq 2^{s-1}-1$. We get $|2x| \in F_{s,t}$ and therefore $2x, -2x \in \pm F_{s,t}$. If $|x| < 2^{2-2^{s-1}}$ then $x \in \pm G_{s,t}$ and there exists $d \in \{0,\dots,2^t-1\}$ with $|x| = d\varepsilon_t 2^{2-2^{s-1}}$. Therefore $|2x| = 2d\varepsilon_t 2^{2-2^{s-1}} \in G_{s,t}$ if $d < 2^{t-1}$ and $|2x| = (2^t + 2(d-2^{t-1}))\varepsilon_t 2^{2-2^{t-1}} = (1+2(d-2^{t-1})\varepsilon_t)2^{2-2^{s-1}} \in F_{s,t}$ if $d \geq 2^{t-1}$. We get $-2x, 2x \in C_{s,t}$. The proof of the second proposition is even easier than the first one: If $|x| \geq 2^{3-2^{s-1}}$, then there exist $d \in \{0,\dots,2^t-1\}, e \in \{3-2^{s-1},\dots,2^{s-1}-1\}$ with $|x| = (1+d\varepsilon_t)2^e$. Therefore $|x|/2 = (1+d\varepsilon_t)2^{e-1}$, with $e-1 \geq 2-2^{s-1}$. We get $|x|/2 \in F_{s,t}$ and therefore $x/2, -x/2 \in \pm F_{s,t}$. □

The next one is a somewhat crude but sometimes helpful estimate.

Lemma 2.22. *Let $s,t \in \mathbb{N}$ and $x \in C_{s,t} \cap K$ with $0 < \mathrm{rd}_{s,t}(|x|)$ and $\mathrm{ru}_{s,t}(|x|) < \infty$. Then*

$$\mathrm{ru}_{s,t}(|x|) \leq 2\mathrm{rd}_{s,t}(|x|)$$

Proof. Let $k \in \mathbb{Z}$ with $|x| \in [2^{k-1}, 2^k]$. Then $2^{k-1} \leq \mathrm{rd}_{s,t}(|x|)$ and therefore $\mathrm{ru}_{s,t}(|x|) \leq 2^k \leq 2\mathrm{rd}_{s,t}(|x|)$. □

Corollary 2.23. *Let $s,t \in \mathbb{N}, x,y \in C_{s,t} \cap K$ and $* \in \{+,-,\cdot,/\}$ with $0 < \mathrm{rd}_{s,t}(|x*y|)$ and $\mathrm{ru}_{s,t}(|x*y|) < \infty$. If $x*y > 0$ we get*

$$x\overline{\circledast}_{s,t}y \leq 2(x\underline{\circledast}_{s,t}y)$$

If $x*y < 0$ we get

$$x\underline{\circledast}_{s,t}y \geq 2(x\overline{\circledast}_{s,t}y)$$

Proof. The proof directly follows from the Lemma 2.22. □

Corollary 2.24. *Let $s,t \in \mathbb{N}$ and $x \in C_{s,t} \cap K$. If $2^{2-2^{s-1}} \leq |x|$ and $\mathrm{ru}_{s,t}(|x|) < 2^{2^{s-1}-1}$, then $\mathrm{rd}_{s,t}(2x) = 2\mathrm{rd}_{s,t}(x)$ and $\mathrm{ru}_{s,t}(2x) = 2\mathrm{ru}_{s,t}(x)$.*
If $2^{3-2^{s-1}} \leq |x|$ and $\mathrm{ru}_{s,t}(|x|) < 2^{2^{s-1}}$, then $\mathrm{rd}_{s,t}(x/2) = \mathrm{rd}_{s,t}(x)/2$ and $\mathrm{ru}_{s,t}(x/2) = \mathrm{ru}_{s,t}(x)/2$.

Proof. Without loss of generality we assume $x > 0$. We have $\mathrm{rd}_{s,t}(x) \leq \mathrm{ru}_{s,t}(x) < 2^{2^{s-1}-1}$. From Lemma 2.21 we get $2\mathrm{rd}_{s,t}(x), 2\mathrm{ru}_{s,t}(x) \in C_{s,t}$. We further have $2\mathrm{rd}_{s,t}(x) \leq 2x \leq 2\mathrm{ru}_{s,t}(x)$. Therefore we get $2\mathrm{rd}_{s,t}(x) \leq \mathrm{rd}_{s,t}(2x)$ and $2\mathrm{ru}_{s,t}(x) \geq \mathrm{ru}_{s,t}(2x)$. We have $2x \geq 2^{3-2^{s-1}}$ and therefore $\mathrm{ru}_{s,t}(2x) \geq \mathrm{rd}_{s,t}(2x) \geq 2^{3-2^{s-1}}$ and therefore with Lemma 2.21 we get $\mathrm{rd}_{s,t}(2x)/2, \mathrm{ru}_{s,t}(2x)/2 \in C_{s,t}$. We further have $\mathrm{rd}_{s,t}(2x)/2 \leq x \leq \mathrm{ru}_{s,t}(2x)/2$. Therefore we get $\mathrm{rd}_{s,t}(2x)/2 \leq \mathrm{rd}_{s,t}(x)$ and $\mathrm{ru}_{s,t}(2x)/2 \geq \mathrm{ru}_{s,t}(x)$.

The second proposition follows from the first one with $x/2$ instead of x, because under the stated conditions from $x/2 \leq \mathrm{ru}_{s,t}(x)/2$ follows $\mathrm{ru}_{s,t}(x/2) \leq \mathrm{ru}_{s,t}(x)/2 < 2^{2^{s-1}-1}$. □

The next corollary states that in $C_{s,t}$ the operations $\overline{\odot}, \underline{\odot}, \overline{\oslash}, \underline{\oslash}$ work very well together with the multiplication and the division by 2.

Corollary 2.25. *Let $s,t \in \mathbb{N}$ and $x,y \in C_{s,t} \cap K$.*
If $2^{2-2^{s-1}} \leq |xy|$ and $\mathrm{ru}(|xy|) < 2^{2^{s-1}-1}$, then

$$(2x)\overline{\odot}_{s,t}y = x\overline{\odot}_{s,t}(2y) = 2(x\overline{\odot}_{s,t}y)$$

$$(2x)\underline{\odot}_{s,t}y = x\underline{\odot}_{s,t}(2y) = 2(x\underline{\odot}_{s,t}y)$$

If $2^{3-2^{s-1}} \leq |xy|$ and $\mathrm{ru}(|xy|) < 2^{2^{s-1}}$, then

$$(x/2)\overline{\odot}_{s,t}y = x\overline{\odot}_{s,t}(y/2) = (x\overline{\odot}_{s,t}y)/2$$

$$(x/2)\underline{\odot}_{s,t}y = x\underline{\odot}_{s,t}(y/2) = (x\underline{\odot}_{s,t}y)/2$$

If $2^{2-2^{s-1}} \leq |x/y|$ and $\mathrm{ru}(|x/y|) < 2^{2^{s-1}-1}$, then

$$(2x)\overline{\oslash}_{s,t}y = x\overline{\oslash}_{s,t}(y/2) = 2(x\overline{\oslash}_{s,t}y)$$

$$(2x)\underline{\oslash}_{s,t}y = x\underline{\oslash}_{s,t}(y/2) = 2(x\underline{\oslash}_{s,t}y)$$

If $2^{3-2^{s-1}} \leq |x/y|$ and $\mathrm{ru}(|x/y|) < 2^{2^{s-1}}$, then

$$x\overline{\oslash}_{s,t}(2y) = (x/2)\overline{\oslash}_{s,t}y = (x\overline{\oslash}_{s,t}y)/2$$

$$x\underline{\oslash}_{s,t}(2y) = (x/2)\underline{\oslash}_{s,t}y = (x\underline{\oslash}_{s,t}y)/2$$

Proof. The proof of this corollary directly follows from Corollary 2.24 □

Lemma 2.26. *Let $s,t \in \mathbb{N}$, $k \in \mathbb{Z}$ with $2^k \in \mathrm{F}_{s,t}$. Let $s_0 := 2^k$ and $s_j := s_{j-1} \oplus_{s,t} 2^{k-j}$ for $j \in \{1,\ldots,t\}$. Then for $j \in \{0,\ldots,t\}$ we have $s_j = \sum_{i=0}^{j} 2^{k-i}$ and for every $\alpha \in C_{s,t}$ with $\alpha \leq 2^{k-t}$ we have $s_t \oplus_{s,t} \alpha \leq 2^{k+1}$. Furthermore, for every $\beta \in C_{s,t}$ with $0 \leq \beta < 2^{k-t}$ we have $2^{k+1} \oplus_{s,t} \beta = 2^{k+1}$.*

Proof. For $j \in \{0,\ldots,t\}$ we have $\sum_{i=0}^{j} 2^{k-i} = 2^k(1 + \varepsilon_t \sum_{i=1}^{j} 2^{t-i}) \in \mathrm{F}_{s,t}$ and thus via induction we get $s_j = \sum_{i=0}^{j} 2^{k-i}$. From $s_t + 2^{k-t} = 2^{k+1}$ we get $s_t \oplus 2^{k-t} = 2^{k+1}$. □

Lemma 2.27. *Let $s,t \in \mathbb{N}$, $\alpha \in \mathrm{F}_{s,t}$ and $k \in \mathbb{Z}$ with $2^k \in C_{s,t}$ and $2^k < \alpha$. Then*

(i) $\alpha \oplus_{s,t} (-2^k) \geq \alpha - 2^{k+1}$

(ii) If $2^{k+t+1} \geq \alpha$ then $\alpha - 2^k \in C_{s,t}$ and therefore $\alpha \oplus_{s,t} (-2^k) = \alpha - 2^k$

Proof. Let $k_1 \in \mathbb{Z}$ with $2^{k_1} \leq \alpha < 2^{k_1+1}$ and $d \in \{0, \ldots, 2^t - 1\}$ with $\alpha = 2^{k_1}(1 + d2^{-t})$. We first show that (ii) implies (i). In case of $2^{k+t+1} \geq \alpha$ this is obvious. In case of $2^{k+t+1} < \alpha$ we have $k + t + 1 \leq k_1$. If $k + t + 1 = k_1$ then $2^{k+t+2} \geq \alpha$ and hence with (ii) we get

$$\alpha \oplus_{s,t} (-2^k) \geq \alpha \oplus_{s,t} (-2^{k+1}) = \alpha - 2^{k+1}$$

If $k + t + 1 < k_1$ we have $\alpha \oplus_{s,t} (-2^k) = \alpha \geq \alpha - 2^{k+1}$. Now we prove (ii). Let $2^{k+t+1} \geq \alpha$. From $2^k < \alpha$ we get $k \leq k_1$ and from $2^{k+t+1} \geq \alpha$ we get $k+t+1 \geq k_1$. In case of $k+t+1 = k_1$ we have $\alpha = 2^{k_1}$ and hence

$$\alpha - 2^k = 2^{k_1} - 2^k = 2^{k_1-1}(1 + (2^t - 1)2^{-t}) \in C_{s,t}$$

In case of $k = k_1$, with $i \in \{0, \ldots, t\}, \tilde{d} \in \{0, \ldots, 2^i - 1\}$ with $d = 2^i + \tilde{d}$ we have

$$\alpha - 2^k = d2^{k_1-t} = 2^{k_1-t+i} + \tilde{d}2^{k_1-t} = 2^{k_1-t+i}(1 + \tilde{d}2^{t-i}2^{-t})$$

which yields $\alpha - 2^k \in C_{s,t}$. In case of $k + t + 1 > k_1 > k$ we have $2^{k-k_1+t} \in \{1, \ldots, 2^{t-1}\}$. We have the equations

$$\alpha - 2^k = 2^{k_1}(1 + d2^{-t}) - 2^k = 2^{k_1}(1 + (d - 2^{k-k_1+t})2^{-t})$$

and

$$\alpha - 2^k = 2^{k_1-1} + 2^{k_1-1} + d2^{k_1-t} - 2^k = 2^{k_1-1}(1 + (2^t + 2(d - 2^{k-k_1+t}))2^{-t})$$

Hence, if $d - 2^{k-k_1+t} \geq 0$ we have $\alpha - 2^k \in C_{s,t}$ with $\alpha - 2^k \in [2^{k_1}, 2^{k_1+1}[$ and if $d - 2^{k-k_1+t} < 0$ we have $\alpha - 2^k \in C_{s,t}$ with $\alpha - 2^k \in [2^{k_1-1}, 2^{k_1}[$. □

2.3 Analysis of error propagation by standard functions

In this section we always assume that M is a finite subset of K and $R \subseteq K \setminus \{0\}, u \in [0, 1[$. Let $C := \overline{M} \cup \{\mathrm{NaN}\}$ and for $* \in \{+, -, \cdot, /\}$ let $\circledast : C^2 \to \overline{M} \cup \{\mathrm{NaN}\}$ functions that fulfill the condition

$$x \circledast_M y \in K \text{ and } \mathrm{e_{rel}}(x * y, x \circledast_M y) \leq u$$

for $x, y \in K \cap C$ with $x * y \in R$.

Now we examine how certain standard Operations, such as summation or multiplication, propagate errors from the operands to the result. Before we state the easy case of summation in Lemma 2.31, we begin with two lemmas about the relative error $\mathrm{e_{rel}}$ and one lemma about products.

Lemma 2.28. *Let $x, y \in K$ and $c \in [0, \infty[$. Then we have the follwing two implications*

$$\mathrm{e_{rel}}(x, y) \leq c < 1 \Rightarrow \mathrm{e_{rel}}(y, x) \leq c/(1 - c)$$
$$\mathrm{e_{rel}}(x, y) \geq c \Rightarrow \mathrm{e_{rel}}(y, x) \geq c/(1 + c)$$

Proof. In case of $x = y = 0$ the implications obviously are true. In case of $x \neq 0$ or $y \neq 0$ we have

$$\mathrm{e}_{\mathrm{rel}}(x, y) = |1 - \frac{y}{x}| \geq 1 - |\frac{y}{x}|$$

and hence

$$\mathrm{e}_{\mathrm{rel}}(y, x) = \mathrm{e}_{\mathrm{rel}}(x, y)\frac{|x|}{|y|} \leq \frac{\mathrm{e}_{\mathrm{rel}}(x, y)}{1 - \mathrm{e}_{\mathrm{rel}}(x, y)}$$

if $\mathrm{e}_{\mathrm{rel}}(x, y) < 1$, which yields the first implication. The proof of the second implication is analogousl. □

Lemma 2.29. *Let $x, y \in K$ and $c \in [0, \infty[$ with $\mathrm{e}_{\mathrm{rel}}(x, y) \leq c$. Then we have $|x| \geq |y|/(1 + c)$ and if $c \in [0, 1]$ we have $|x| \leq |y|/(1 - c)$.*

Proof. We have $|1 - y/x| \leq c$ and hence $|y/x| = y/x \in [1 - c, 1 + c]$. From that, we get the proposed inequalities. □

Lemma 2.30. *Let $c_1, \ldots, c_n \in K$ and $\delta_1, \ldots, \delta_n \in K$ mit $|\delta_1| \leq c_1, \ldots, |\delta_n| \leq c_n$. Then*

$$\left|\prod_{i=1}^{n}(1 + \delta_i) - 1\right| \leq \prod_{i=1}^{n}(1 + c_i) - 1$$

Proof.

$$\left|\prod_{i=1}^{n}(1 + \delta_i) - 1\right| = \left|\sum_{k=0}^{n}\sum_{\substack{I\subseteq\{1,\ldots,n\}\\|I|=k}}\prod_{i\in I}\delta_i - 1\right| \leq \sum_{k=1}^{n}\sum_{\substack{I\subseteq\{1,\ldots,n\}\\|I|=k}}\prod_{i\in I}c_i = \prod_{i=1}^{n}(1 + c_i) - 1$$

□

Lemma 2.31. *Let $x, y \in K$, $\tilde{x}, \tilde{y} \in K \cap C$ and $c_1, c_2 \in [0, \infty[$ with*

$$\mathrm{e}_{\mathrm{rel}}(x, \tilde{x}) \leq c_1, \mathrm{e}_{\mathrm{rel}}(y, \tilde{y}) \leq c_2$$

If $\tilde{x} - \tilde{y} \in R$, then

$$\mathrm{e}_{\mathrm{rel}}(x - y, \tilde{x} \ominus \tilde{y}) \leq (1 + u)\left(1 + \frac{c_1|x| + c_2|y|}{|x - y|}\right) - 1$$

Proof. Let $\delta_1 := \frac{x-\tilde{x}}{x}$, $\delta_2 := \frac{y-\tilde{y}}{y}$ and $\eta := \frac{\tilde{x}-\tilde{y}-(\tilde{x}\ominus\tilde{y})}{\tilde{x}-\tilde{y}}$. Then

$$\tilde{x} \ominus \tilde{y} = (\tilde{x} - \tilde{y})(1 - \eta) = (x(1 - \delta_1) - y(1 - \delta_2))(1 - \eta)$$

It follows

$$\begin{aligned} |\tilde{x} \ominus \tilde{y} - (x - y)| &= |(1 - \eta)(-x\delta_1 + y\delta_2) - \eta(x - y)| \\ &\leq (1 + u)(c_1|x| + c_2|y|) + u|x - y| \end{aligned}$$

and thus the conclusion. □

We state another lemma on error propagation in a subtraction. Before that, we state the following easy consequence of the triangle inequality.

Lemma 2.32. *Let $x, y, \varepsilon, \delta \in K$ with $|1-x| \leq \varepsilon$ and $|1-y| \geq \delta$. Then $|1-xy| \geq (1+\delta)(1-\varepsilon) - 1$.*

Proof. Without loss of generality we assume $\delta \geq 0$. We have $\varepsilon \geq 0$. We assume $\varepsilon < 1$ and $\delta > \varepsilon/(1-\varepsilon)$, otherwise it would be $(1+\delta)(1-\varepsilon)-1 \leq 0$ and hence the proposition obviously be true. Especially we have $\delta \geq \varepsilon$ and $x > 0$. In case of $y \geq 1$ we have $y \geq 1+\delta \geq 1+\varepsilon/(1-\varepsilon)$ and therefore $xy \geq (1-\varepsilon)y \geq 1$ and hence $|1-xy| = xy - 1 \geq (1-\varepsilon)(1+\delta) - 1$. In case of $0 < y < 1$ we have $xy \leq (1+\varepsilon)(1-\delta) = 1 + \varepsilon - \delta - \varepsilon\delta \leq 1$ and therefore $|1-xy| = 1 - xy \geq 1 - (1+\varepsilon)(1-\delta) \geq (1-\varepsilon)(1+\delta) - 1$. In case of $y \leq 0$ we have $|1-xy| = 1 - xy \geq 1 - (1-\varepsilon)(1-\delta) \geq (1+\delta)(1-\varepsilon) - 1$. □

Now we state another lemma on the error propagation which can occur when a subtraction $\ominus$ is performed.

Lemma 2.33. *Let $x, y \in K$, $\tilde{x}, \tilde{y} \in K \cap C, c_1, c_2, c_3 \in [0, \infty[$ with $\mathrm{e}_{\mathrm{rel}}(x, \tilde{x}) \leq c_1$, $\mathrm{e}_{\mathrm{rel}}(y, \tilde{y}) \leq c_2 < 1$, $\mathrm{e}_{\mathrm{rel}}(x, \tilde{y}) \geq c_3$. Let $d := \frac{c_1}{(1-c_2)(1+c_3/(1+c_3))-1} + \frac{c_2}{(1+c_3)(1-c_2/(1-c_2))-1}$. If $\tilde{x} - \tilde{y} \in R$ and $(1-c_2)(1+c_3/(1+c_3)), (1+c_3)(1-c_2/(1-c_2)) > 1$, then $x \neq y$ and*

$$\mathrm{e}_{\mathrm{rel}}(x-y, \tilde{x} \ominus \tilde{y}) \leq (1+u)(1+d) - 1$$

Proof. We have $|1 - \tilde{y}/x| \geq c_3$ and $|1 - \tilde{y}/y| \leq c_2$. Because of Lemma 2.28 we further have $|1 - x/\tilde{y}| \geq c_3/(1+c_3)$ and $|1 - y/\tilde{y}| \leq c_2/(1-c_2)$. Thus, by Lemma 2.32 we get

$$\mathrm{e}_{\mathrm{rel}}(y, x) = |1 - (x/\tilde{y}) \cdot (\tilde{y}/y)| \geq (1-c_2)(1+c_3/(1+c_3)) - 1$$

$$\mathrm{e}_{\mathrm{rel}}(x, y) = |1 - (y/\tilde{y}) \cdot (\tilde{y}/x)| \geq (1+c_3)(1-c_2/(1-c_2)) - 1$$

By Lemma 2.31 we get

$$\mathrm{e}_{\mathrm{rel}}(x-y, \tilde{x} \ominus \tilde{y}) \leq (1+u)(1 + c_1/\mathrm{e}_{\mathrm{rel}}(y, x) + c_2/\mathrm{e}_{\mathrm{rel}}(x, y)) - 1$$

and thus the conclusion. □

Lemma 2.34. *Let $m \in \mathbb{N}$, $x_1, \ldots, x_m \in K$, $y_1, \ldots, y_m \in K \cap C$, $c_1, \ldots, c_m \in [0, \infty[$ with*

$$\mathrm{e}_{\mathrm{rel}}(x_1, y_1) \leq c_1, \ldots, \mathrm{e}_{\mathrm{rel}}(x_m, y_m) \leq c_m$$

and

$$q_1 := y_1, q_i := q_{i-1} \odot y_i \text{ for } i \in \{2, \ldots, m\}$$

If $q_{i-1} y_i \in R$ for $i \in \{2, \ldots, m\}$, then

$$\mathrm{e}_{\mathrm{rel}}\left(\prod_{i=1}^{m} x_i, q_m\right) \leq (1+u)^{m-1} \prod_{i=1}^{m} (1+c_i) - 1$$

Proof. Let $\delta_1 := \frac{x_1 - y_1}{x_1}, \ldots, \delta_m := \frac{x_m - y_m}{x_m}$, $\varepsilon_1 := 0$ and $\varepsilon_i := \frac{q_{i-1}y_i - q_i}{q_{i-1}y_i}$ for $i \in \{2, \ldots, m\}$. Then

$$q_1 = y_1 = x_1(1 - \delta_1) = x_1(1 - \delta_1)(1 - \varepsilon_1)$$

and

$$q_i = q_{i-1}y_i(1 - \varepsilon_i) = q_{i-1}x_i(1 - \varepsilon_i)(1 - \delta_i)$$

for $i \in \{2, \ldots, m\}$. Therefore, for $j \in \{1, \ldots, m\}$ we have

$$q_j = \prod_{i=1}^{j} x_i(1 - \varepsilon_i)(1 - \delta_i)$$

Hence, with the use of lemma 2.30 we get

$$\begin{aligned} \left| q_m - \prod_{i=1}^{m} x_i \right| &= \left| \prod_{i=1}^{m} x_i \right| \left| \prod_{i=1}^{m} (1 - \varepsilon_i)(1 - \delta_i) - 1 \right| \\ &\leq \left| \prod_{i=1}^{m} x_i \right| \left((1 + u)^{m-1} \prod_{i=1}^{m} (1 + c_i) - 1 \right) \end{aligned}$$

which yields the proposition. □

Lemma 2.35. *Let $x \in K, y \in K \setminus \{0\}, \tilde{x}, \tilde{y} \in K \cap C$ with $\tilde{x}/\tilde{y} \in R$ and $c_1 \in [0, \infty[, c_2 \in [0, 1[$ with*

$$\mathrm{e}_{\mathrm{rel}}(x, \tilde{x}) \leq c_1, \mathrm{e}_{\mathrm{rel}}(y, \tilde{y}) \leq c_2$$

Then

$$\mathrm{e}_{\mathrm{rel}}(x/y, \tilde{x} \oslash \tilde{y}) \leq (1 + c_1)(1 + u)/(1 - c_2) - 1$$

Proof. Let $\delta_1 := \frac{x - \tilde{x}}{x}$, $\delta_2 := \frac{y - \tilde{y}}{y}$ and $\eta := \frac{\tilde{x}/\tilde{y} - \tilde{x} \oslash \tilde{y}}{\tilde{x}/\tilde{y}}$. We have

$$\begin{aligned} \tilde{x} \oslash \tilde{y} &= \tilde{x}/\tilde{y}(1 - \eta) \\ &= x/y(1 - \delta_1)/(1 - \delta_2)(1 - \eta) \\ &= x/y(1 - \delta_1)(1 + \frac{\delta_2}{1 - \delta_2})(1 - \eta) \end{aligned}$$

With lemma 2.30 we get

$$|x/y - \tilde{x} \oslash \tilde{y}| \leq |x/y| \left((1 + c_1)(1 + \frac{c_2}{1 - c_2})(1 + u) - 1 \right) = |x/y|((1 + c_1)(1 + u)/(1 - c_2) - 1)$$

and thus the proposed inequality. □

Theorem 2.36. *Let $m \in \mathbb{N}$, $c \in [0, \infty[$ and $x_1, \ldots, x_m \in K$, $y_1, \ldots, y_m \in K \cap C$ with*

$$\mathrm{e}_{\mathrm{rel}}(x_1, y_1), \ldots, \mathrm{e}_{\mathrm{rel}}(x_m, y_m) \leq c$$

and $s_1 := y_1$ and $s_i := s_{i-1} \oplus y_i$ for $i \in \{2, \ldots, m\}$. If $s_{i-1} + y_i \in R$ for $i \in \{2, \ldots, m\}$, then

$$|s_m - \sum_{i=1}^{m} x_i| \quad \leq \quad ((1+c)(1+u)^{m-1} - 1) \sum_{i=1}^{m} |x_i|$$

Especially, if $x_i \geq 0$ for every $i \in \{1, \ldots, m\}$ or if $x_i \leq 0$ for every $i \in \{1, \ldots, m\}$:

$$\mathrm{e}_{rel}\left(\sum_{i=1}^{m} x_i, s_m\right) \leq (1+c)(1+u)^{m-1} - 1$$

Proof. Let $\delta_1 := \frac{x_1 - y_1}{x_1}, \ldots, \delta_m := \frac{x_m - y_m}{x_m}$ and $\varepsilon_1 := 0$ and $\varepsilon_i := \frac{s_{i-1} + y_i - s_i}{s_{i-1} + y_i}$ for $i \in \{2, \ldots, m\}$. We have

$$s_1 = x_1(1 - \delta_1)(1 - \varepsilon_1)$$

and for $i \in \{2, \ldots, m\}$

$$s_i = (s_{i-1} + x_i(1 - \delta_i))(1 - \varepsilon_i)$$

and hence for $j \in \{1, \ldots, m\}$

$$s_j = \sum_{i=1}^{j} \left(x_i(1 - \delta_i) \prod_{\ell=i}^{j} (1 - \varepsilon_\ell) \right)$$

Hence, with $\vartheta_i := (1 - \delta_i) \prod_{\ell=i}^{m} (1 - \varepsilon_\ell) - 1$ we have

$$\begin{aligned} |s_m - \sum_{i=1}^{m} x_i| \quad &\leq \quad \sum_{i=1}^{m} |x_i \vartheta_i| \\ &\leq \quad |x_1|((1+c)(1+u)^{m-1} - 1) + \sum_{i=2}^{m} |x_i|((1+c)(1+u)^{m-i+1} - 1) \\ &\leq \quad ((1+c)(1+u)^{m-1} - 1) \sum_{i=1}^{m} |x_i| \end{aligned}$$

□

Lemma 2.37. *Let $x, y \in K$, $\tilde{x}, \tilde{y} \in K \cap C$, $c_1, c_2 \in [0, \infty[$ with*

$$\mathrm{e}_{\mathrm{rel}}(x, \tilde{x}) \leq c_1, \mathrm{e}_{\mathrm{rel}}(y, \tilde{y}) \leq c_2$$

Let $\tilde{x} + \tilde{y} \in R$. Then

$$\mathrm{e}_{\mathrm{rel}}(x + y, \tilde{x} \oplus \tilde{y}) \leq (1+u)\left(1 + \frac{c_1|x| + c_2|y|}{|x+y|}\right) - 1$$

Proof. Let $\varepsilon_1 := \frac{x-\tilde{x}}{x}$ and $\varepsilon_2 := \frac{y-\tilde{y}}{y}$. Hence $\tilde{x} = x(1-\varepsilon_1)$, $\tilde{y} = y(1-\varepsilon_2)$ and $|\varepsilon_1| \leq c_1$, $|\varepsilon_2| \leq c_2$. Let $\delta := \frac{\tilde{x}+\tilde{y}-\tilde{x}\oplus\tilde{y}}{\tilde{x}+\tilde{y}}$. Then

$$\tilde{x} \oplus \tilde{y} = (\tilde{x} + \tilde{y}) \cdot (1-\delta) = x(1-\varepsilon_1)(1-\delta) + y(1-\varepsilon_2)(1-\delta)$$

and with that we get

$$\begin{aligned} & |x + y - \tilde{x} \oplus \tilde{y}| \\ = \; & |x + y - (x(1-\varepsilon_1)(1-\delta) + y(1-\varepsilon_2)(1-\delta))| \\ = \; & |(1-\delta)(\varepsilon_1 x + \varepsilon_2 y) + \delta(x+y)| \\ \leq \; & (1+u)(c_1|x| + c_2|y|) + u|x+y| \end{aligned}$$

Hence

$$\begin{aligned} \mathrm{e}_{\mathrm{rel}}(x+y, \tilde{x} \oplus \tilde{y}) &= \left| \frac{x + y - \tilde{x} \oplus \tilde{y}}{x+y} \right| \\ &\leq (1+u)\frac{c_1|x| + c_2|y|}{|x+y|} + u \end{aligned}$$

which yields the proposition. □

Now, we state a lemma on the error propagation in a summation where one of the summands has a much larger absolute value than the other:

Lemma 2.38. *Let $x, y \in K$, $\tilde{x}, \tilde{y} \in K \cap C$, $c_1, c_2 \in [0, \infty[$ with*

$$\mathrm{e}_{\mathrm{rel}}(x, \tilde{x}) \leq c_1, \mathrm{e}_{\mathrm{rel}}(y, \tilde{y}) \leq c_2$$

Let $c_3 \in]1, \infty[$ with $|x|/|y| \geq c_3$. Let $\tilde{x} + \tilde{y} \in R$. Then

$$\mathrm{e}_{\mathrm{rel}}(x+y, \tilde{x} \oplus \tilde{y}) \leq (1+u)\left(1 + \frac{c_1}{1 - c_3^{-1}} + \frac{c_2}{c_3 - 1}\right) - 1$$

Proof. We have the following inequatlitiy

$$\begin{aligned} \mathrm{e}_{\mathrm{rel}}(x+y, \tilde{x} \oplus \tilde{y}) &\leq (1+u)\left(1 + \frac{c_1|x| + c_2|y|}{|x+y|}\right) - 1 \\ &= (1+u)\left(1 + \frac{c_1}{|1 + y/x|} + \frac{c_2}{|x/y + 1|}\right) - 1 \\ &\leq (1+u)\left(1 + \frac{c_1}{1 - |y/x|} + \frac{c_2}{|x/y| - 1}\right) - 1 \\ &\leq (1+u)\left(1 + \frac{c_1}{1 - c_3^{-1}} + \frac{c_2}{c_3 - 1}\right) - 1 \end{aligned}$$

□

The next Lemma states an error bound for summation in cases where the absolute value of some of the summands is smaller than a certain bound δ. For example, δ could be the smallest positive normal machine number.

Lemma 2.39. *Let $n \in \mathbb{N}$, $a_1 \ldots, a_n \in K, b_1, \ldots, b_n \in K \cap C$. Let $c, \delta \in [0, \infty[$, $m \in \{0, \ldots, n\}$ with $\mathrm{e}_{\mathrm{rel}}(a_k, b_k) \leq c$ for $k \in \{1, \ldots, m\}$ and $|a_k|, |b_k| \leq \delta$ for $k \in \{m+1, \ldots, n\}$. Let $s_1 := b_1$ and $s_k := s_{k-1} \oplus b_k$ for $k \in \{2, \ldots, n\}$. We assume $s_{k-1} + b_k \in R$ for $k \in \{2, \ldots, n\}$. Then*

$$\left| \sum_{k=1}^{n} a_k - s_n \right| \leq \sum_{k=1}^{n} |a_k|((1+c)(1+u)^{n-1} - 1) + 2n\delta(1+u)^{n-1}$$

Proof. Let $\delta_1 := \frac{a_1 - b_1}{a_1}; \ldots, \delta_m := \frac{a_m - b_m}{a_m}$, so that

$$b_1 = a_1(1-\delta_1), \ldots, b_m = a_m(1-\delta_m)$$

and $\varepsilon_1 := 0$, $\varepsilon_2 := \frac{s_1 + b_2 - s_2}{s_1 + b_2}, \ldots, \varepsilon_n := \frac{s_{n-1} + b_n - s_n}{s_{n-1} + b_n}$, so that $s_1 = b_1(1-\varepsilon_1)$ and $s_k = (s_{k-1} + b_k)(1-\varepsilon_k)$ for $k \in \{2, \ldots, n\}$. By induction we get

$$s_n \;=\; \sum_{j=1}^{n} b_j \prod_{k=j}^{n} (1-\varepsilon_k)$$

and with that

$$\begin{aligned}
& \left| \sum_{j=1}^{n} a_j - s_n \right| \\
= \; & \left| \sum_{j=1}^{m} a_j (1 - (1-\delta_j) \prod_{k=j}^{n} (1-\varepsilon_k)) + \sum_{j=m+1}^{n} a_j - \sum_{j=m+1}^{n} b_j \prod_{k=j}^{n} (1-\varepsilon_k) \right| \\
\leq \; & \sum_{j=1}^{m} |a_j|((1+c)(1+u)^{n-1} - 1) + \sum_{j=m+1}^{n} \underbrace{\left| a_j - b_j \prod_{k=j}^{n} (1-\varepsilon_k) \right|}_{\leq 2\delta(1+u)^{n-1}} \\
\leq \; & \sum_{j=1}^{n} |a_j|((1+c)(1+u)^{n-1} - 1) + 2n\delta(1+u)^{n-1}
\end{aligned}$$

and hence the proposition. □

Now we use Lemma 2.39 to approximate sums of the form $a + b\sum_{k=1}^{n} \frac{x^k}{2k+1}$. Here some of the summands may be smaller than the smallest positive normal machine number and hence may not be computed with small relative error.

Lemma 2.40. *Let $n \in \mathbb{N}$, $k_0 \in \{1, \ldots, n\}$, $a, b, x \in K$, $c_{k_0}, \ldots, c_n \in K \setminus \{0\}$ and $s_{k_0-1}, \tilde{x}_0, \tilde{x}$, $\tilde{c}_{k_0}, \ldots, \tilde{c}_n \in K \cap C$. Let $K_1, K_2, K_3 \in [0, \infty[$, $K_4 \in [0, 1[$ with*

$$\mathrm{e}_{\mathrm{rel}}(b, \tilde{x}_0) \leq K_1, \mathrm{e}_{\mathrm{rel}}(x, \tilde{x}) \leq K_2, \mathrm{e}_{\mathrm{rel}}(a, s_{k_0-1}) \leq K_3$$

and $\mathrm{e}_{\mathrm{rel}}(c_k, \tilde{c}_k) \leq K_4$ for $k \in \{k_0, \ldots, n\}$. For $k \in \{1, \ldots, n\}$ let $\tilde{x}_k := \tilde{x}_{k-1} \odot \tilde{x}$. For $k \in \{k_0, \ldots, n\}$ let $z_k := \tilde{x}_k \oslash \tilde{c}_k$ and $s_k := s_{k-1} \oplus z_k$. Let $\delta \in [0, \infty[$ and $m \in \{0, \ldots, n\}$ with $\tilde{x}_{k-1} \cdot \tilde{x} \in R$ for $k \in \{1, \ldots, m\}$, $\tilde{x}_k / \tilde{c}_k \in R$ for $k \in \{k_0, \ldots, m\}$ and $|z_k|, |bx^k/c_k| \leq \delta$ for $k \in \{k_0, \ldots, n\}$ with $k > m$. Let $s_{k-1} + z_k \in R$ for $k \in \{k_0, \ldots, n\}$. Then, with $K_5 := \max(K_3, (1 + K_1)(1 + K_2)^n(1 + u)^{n+1}/(1 - K_4) - 1)$ we have

$$\left| a + b \sum_{k=k_0}^{n} \frac{x^k}{c_k} - s_n \right| \leq$$
$$\left(|a| + |b| \sum_{k=k_0}^{n} \frac{|x|^k}{|c_k|} \right) ((1 + K_5)(1 + u)^{n-k_0+1} - 1) + 2(n - k_0 + 2)\delta(1 + u)^{n-k_0+1}$$

Proof. From Lemmas 2.34 and 2.35 we get

$$\mathrm{e}_{\mathrm{rel}}(bx^k, \tilde{x}_k) \leq (1 + K_1)(1 + K_2)^k(1 + u)^k - 1$$

for $k \in \{1, \ldots, m\}$ and

$$\begin{aligned} \mathrm{e}_{\mathrm{rel}}(b\frac{x^k}{c_k}, z_k) &\leq (1 + K_1)(1 + K_2)^k(1 + u)^{k+1}/(1 - K_4) - 1 \\ &\leq (1 + K_1)(1 + K_2)^n(1 + u)^{n+1}/(1 - K_4) - 1 \end{aligned}$$

for $k \in \{k_0, \ldots, m\}$. With the use of this bounds, by Lemma 2.39 we get the proposition. □

Chapter 3

Analysis of error propagation in Loader's algorithm for the binomial density

In this chapter, after a review of Loader's algorithm for the binomial density we derive bounds for the error propagation in Loader's algorithm. In the entire chapter we use the the notions we defined in chapter 2. We often will write $\circledast, \underline{\circledast}, \overline{\circledast}$ as abbreviations for $\circledast_{s,t}, \underline{\circledast}_{s,t}, \overline{\circledast}_{s,t}$.

3.1 Loader's algorithm for the binomial density

In this section we assume $K = \mathbb{R}$. An algorithm for computing the binomial density

$$\mathrm{b}_{n,p}(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

is given by Loader [16]. The statistical software R [20] computes the binomial density with a slightly modified version of Loader's algorithm. The version that R uses is stated in Appendix D. The command for executing the algorithm with R is `dbinom(x,n,p)`. In case of $p \in \,]0,1[$, $n \in \{2, \ldots, 2^{53}\}$, $x \in \{1, \ldots, n-1\}$ the following program, which is written in the programming language C according to the C standard defined in [2], is very similar to the version of Loader's algorithm that R uses.

```
double bin(double x, double n, double p){
 double q = 1-p;
 double lc = stirlerr(n)-stirlerr(x)-stirlerr(n-x)-bd0(x,n*p)-bd0(n-x,n*q);
 double lf = M_LN_2PI + log(x) + log1p(- x/n);
 return exp(lc - 0.5*lf);
}
```

Here `M_LN_2PI` is the element of the set of IEEE-Double numbers which is closest to $\log(2\pi)$, the function `log1p` is an approximation for $]-1, \infty[\ni x \mapsto \log(1+x)$, the function `stirlerr`

is an approximation for $]0,\infty[\ni x \mapsto S(x) := \log(\Gamma(x+1)/(x^x \mathrm{e}^{-x}\sqrt{2\pi x}))$ and the function `bd0` is an approximation for $]0,\infty[^2 \ni (x,np) \mapsto x\log(x/np) - x + np$. The C programs that define the functions `stirlerr` and `bd0` are displayed in appendix D. A brief summary of analytical approximations of $S(x)$ can be found in appendix C.

The basic idea of this algorithm can be understood by regarding the equation

$$\mathrm{b}_{n,p}(x) = \mathrm{b}_{n,\frac{x}{n}}(x)\frac{\mathrm{b}_{n,p}(x)}{\mathrm{b}_{n,\frac{x}{n}}(x)}$$

in which the value $\mathrm{b}_{n,x/n}(x)$ is called "saddle point approximator" and the fraction $\mathrm{b}_{n,p}(x)/\mathrm{b}_{n,x/n}(x)$ is called "deviance part". The problem of directly computing $\mathrm{b}_{n,p}(x)$ here has changed to computing saddle point approximator and deviance part first and then getting $\mathrm{b}_{n,p}(x)$ by multiplying them. The advantage of this "saddle point shift" is, that while $\mathrm{b}_{n,p}(x)$ could be very small at the tails of the binomial distribution, the approximator $\mathrm{b}_{n,x/n}(x)$ is not that small because the binomial density $\mathrm{b}_{n,p}$ has its maximal values near its expectation $\mu = np$ and in case of $p = x/n$ its expectation is $\mu = x$. It is assumed that it is easier to compute values of the binomial density accurately which are not very small, than to compute very small values accurately.

The saddle point approximator can be written in the form

$$\mathrm{b}_{n,\frac{x}{n}}(x) = \binom{n}{x}\left(\frac{x}{n}\right)^x\left(\frac{n-x}{n}\right)^{n-x} = \frac{\frac{n!}{n^n}}{\frac{x!}{x^x}\frac{(n-x)!}{(n-x)^{n-x}}} \tag{3.1}$$

Now the idea is, to use Stirling's series to approximate the three minor fractions in the last expression. To do so, the expression first is brought into the form

$$\mathrm{b}_{n,\frac{x}{n}}(x) = \frac{\frac{n!}{n^n\mathrm{e}^{-n}\sqrt{2\pi n}}}{\frac{x!}{x^x\mathrm{e}^{-x}\sqrt{2\pi x}}\cdot\frac{(n-x)!}{(n-x)^{n-x}\mathrm{e}^{-n+x}\sqrt{2\pi(n-x)}}}\cdot\frac{1}{\sqrt{2\pi x\left(1-\frac{x}{n}\right)}} \tag{3.2}$$

where Stirling's approximations can be applied three times. In order to do that in an easy way, the last equation is transformed by the logarithm, which means one turns to compute $\log(\mathrm{b}_{n,p}(x))$ instead of $\mathrm{b}_{n,p}(x)$ and will get $\mathrm{b}_{n,p}(x)$ by exponentiation in the end. The transformation via logarithm yields the equation

$$\begin{aligned}\log\left(\mathrm{b}_{n,\frac{x}{n}}(x)\right) &= \log\left(\frac{n!}{n^n\mathrm{e}^{-n}\sqrt{2\pi n}}\right) - \log\left(\frac{x!}{x^x\mathrm{e}^{-x}\sqrt{2\pi x}}\right)\\ &\quad - \log\left(\frac{(n-x)!}{(n-x)^{n-x}\mathrm{e}^{-n+x}\sqrt{2\pi(n-x)}}\right) - \log\left(\sqrt{2\pi x\left(1-\frac{x}{n}\right)}\right)\end{aligned}$$

The function `stirlerr` in the algorithm stated above is a tool to approximate the first three expressions on the right side of this equation, using Stirling's series. The last of the four expressions on the right side is the one which occurs as the value `-0.5*lf` in the algorithm.

The logarithm of the deviance part is

$$\begin{aligned}\log\left(\frac{\mathrm{b}_{n,p}(x)}{\mathrm{b}_{n,\frac{x}{n}}(x)}\right) &= \log\left(\frac{p^x(1-p)^{n-x}}{(\frac{x}{n})^x(\frac{n-x}{n})^{n-x}}\right)\\ &= x\log\left(\frac{np}{x}\right)+(n-x)\log\left(\frac{n(1-p)}{n-x}\right) \qquad (3.3)\end{aligned}$$

We remark that because of cancelation here no more binomial coefficients occur. For more accurate numerical computation, Loader brought this expression into the form

$$\log\left(\frac{\mathrm{b}_{n,p}(x)}{\mathrm{b}_{n,\frac{x}{n}}(x)}\right) = -f(x,np)-f(n-x,n(1-p)) \qquad (3.4)$$

with $f(x,y) := x\log(x/y)-x+y$ which is the function that is approximated by `bd0` in the algorithm stated above. We remark that because f is positive, which is a direct consequence of the well known inequality $\log(t) \geq 1-1/t$, in (3.4) both terms on the right side are negative whereas in (3.3) one was negative and the other one was positive.

To get a deeper understanding of Loader's algorithm, we briefly want to compare it to a classical method of computing the value $\mathrm{b}_{n,p}(x)$, which was used before Loader's algorithm has been developed. The logarithm of the probability $\mathrm{b}_{n,p}(x)$ has been computed according to the formula

$$\log(\mathrm{b}_{n,p}(x)) = \log(n!)-\log(x!)-\log((n-x)!)+x\log(p)+(n-x)\log(1-p) \qquad (3.5)$$

where $\log(n!)$ could be computed with the help of Stirling's series via

$$\log(n!) = \log\left(\frac{n!}{n^n\mathrm{e}^{-n}\sqrt{2\pi n}}\right)+(n+\frac{1}{2})\log(n)-n+\frac{1}{2}\log(2\pi)$$

and $\log(x!)$ and $\log((n-x)!)$ analogously. In (3.5) the problem is, that the logarithms of the factorials $n!$, $x!$ and $(n-x)!$ typically are very large compared to the absolute value of the result $\log(\mathrm{b}_{n,p}(x))$. Because of this, moderate relative errors in the operands can lead to a very large relative error in the result of the subtraction. This problem does not exist in Loader's algorithm anymore, because in the deviance part as mentioned no more binomial coefficients do occur, which were possibly very large compared to the result, and in the saddle point approximator, in transition from equation (3.1) to (3.2), the possibly large expressions e^x, e^n and e^{n-x} canceled out.

We conclude this overview by looking again at the expression $\mathrm{b}_{n,x/n}(x)$ in equation (3.2). If we approximate the three minor fractions by 1, we get the approximation

$$\mathrm{b}_{n,\frac{x}{n}}(x) \approx \frac{1}{\sqrt{2\pi x\left(1-\frac{x}{n}\right)}}$$

which with $p = x/n$ also can be written as

$$\mathrm{b}_{n,p}(np) \approx \frac{1}{\sqrt{2\pi np(1-p)}} = \varphi_{np,np(1-p)}(np)$$

where $\varphi_{\mu,\sigma^2} = \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(x-\cdot)^2}{2\sigma^2}}$ is the density of the normal distribution with mean μ and variance σ^2. Hence, the way in which Loader's algorithm works can be comprehensed in the following sentence: A saddle point shift is performed where the saddle point approximator is computed by a normal approximation in the center of the corresponding normal distribution and the deviance part should be computed numerically accurate because due to cancelation of binomial coefficients no more subtraction of large operands with small result does occur.

In the remainder of this chapter we will examine the error propagation in Loader's algorithm, assuming that all computations are performed by a machine with machine precision $u \in]0,1[$. That means our aim is the following.

If $p \in]0,1[$ is approximated by a machine number $\tilde{p}$ and the relative error in this approximation is bounded by $c \in]0,\infty[$, we want to derive a bound for the relative error in the approximation of $\mathrm{b}_{n,p}(x)$ by the computed result $\texttt{bin}(x,n,\tilde{p})$, assuming n and x are machine numbers.

From the following example we infer that it can not be possible to derive bounds less than $1/5$ for every value of $p \in]0,1[$, if the computations are performed in the IEEE-Double number system.

Example 3.1. Let $n = 2^{53} = \max\{n \in \mathbb{N} : \{1,\ldots,n\} \subseteq \mathrm{IEEEDouble}\}$, $x = n-1$, $p = 1-2^{-54}$. We compute an approximator for $\mathrm{b}_{n,p}(x)$ with R. We use the value $\tilde{p} = 1-2^{-53}$ as approximator for p. One might imagine that we do not know p and got the approximator $\tilde{p}$ by a numerical computation. The relative error in the approximation of p by $\tilde{p}$ is

$$\mathrm{e}_{\mathrm{rel}}(p,\tilde{p}) = 2^{-54}/(1-2^{-54}) < 2^{-53}$$

Hence, the approximation of p by $\tilde{p}$ is good considering the machine precision $u_{52} = 2^{-53}$ of the IEEE-Double Number System, in which the computations with R are performed. Evaluation with R returns the value dbinom(x,n,$\tilde{p}$) $= 0.3678794$ as approximator for $\mathrm{b}_{n,p}(x)$, but for the exact result with (C.1) and (C.2) from appendix C we get the inequality

$$\begin{aligned}
\log\left(\mathrm{b}_{n,p}(x)\right) &= \log\left(\frac{n!}{n^n\mathrm{e}^{-n}\sqrt{2\pi n}}\right) - \log\left(\frac{x!}{x^x\mathrm{e}^{-x}\sqrt{2\pi x}}\right) \\
&\quad - \log\left(\frac{(n-x)!}{(n-x)^{n-x}\mathrm{e}^{-n+x}\sqrt{2\pi(n-x)}}\right) - \log\left(\sqrt{2\pi x\left(1-\frac{x}{n}\right)}\right) \\
&\quad + x\log\left(\frac{np}{x}\right) + (n-x)\log\left(\frac{n(1-p)}{n-x}\right) \\
&\leq \frac{1}{12n} - (\frac{1}{12x} - \frac{1}{360x^3}) - (\frac{1}{12(n-x)} - \frac{1}{360(n-x)^3}) \\
&\quad - \log\left(\sqrt{2\pi x\left(1-\frac{x}{n}\right)}\right) + x\log\left(\frac{np}{x}\right) + (n-x)\log\left(\frac{n(1-p)}{n-x}\right)
\end{aligned}$$

Computation with the computer algebra system Mathematica verifies that the right side of this inequality is less than $-\frac{1192641}{1000000}$ and that $\exp(-\frac{1192641}{1000000}) < \frac{30342}{100000}$. Hence, provided that these verifications with Mathematica are reliable, we have $\mathrm{b}_{n,p}(x) < \frac{30342}{100000}$ and the relative error in the approximation of $\mathrm{b}_{n,p}(x)$ by the approximator we computed with R must be larger than $1/5$.

3.2 Overview about research on the accuracy of algorithms for the binomial density

Loader [16] performs numerical experiments which indicate numerical accuracy of Loader's algorithm for the binomial density.

Kaiser [12] in examples studied the accuracy of a so called "multiplication method" which was stated in Appendix B of Loader [16]. This method multiplicates all the factors in the representation

$$\mathrm{b}_{n,p}(x) = \prod_{i=1}^{x} \frac{n-x+i}{i} \prod_{i=1}^{x} p \prod_{i=1}^{n-x} (1-p)$$

in an order which aims to prevent numerical underflow. Here, numerical underflow means that a result of a computer operation is smaller than the smallest positive computer number.

Hirai and Nakamura [10] construct a new arithmetic system in the programming language C. They further propose an algorithm for the computation of the binomial density using the proposed arithmetic system. Further Hirai and Nakamura perform numerical experiments which indicate usefulness of the proposed algorithm for a very large range of sample size n.

3.3 Error propagation in the computation of np and $n(1-p)$

We start our examination of error propagation in Loader's algorithm with examining how bounds for the relative error $\mathrm{e}_{\mathrm{rel}}(p, \tilde{p})$ propagate when the values $n \odot \tilde{p}$ and $n \odot (1 \ominus \tilde{p})$ are computed, which in Loader's algorithm occur as inputs for the function `bd0`.

Lemma 3.2. *Let $n \in K \cap C$ and $p \in]0,1[$, $\tilde{p} \in]0,1[\cap C$ and $c \in [0, \infty[$ with*

$$\max\left(\mathrm{e}_{\mathrm{rel}}(p,\tilde{p}), \mathrm{e}_{\mathrm{rel}}(p,\tilde{p})\frac{p}{1-p}\right) \leq c$$

Then

$$\mathrm{e}_{\mathrm{rel}}(np, n \odot \tilde{p}) \leq (1+u)(1+\mathrm{e}_{\mathrm{rel}}(p,\tilde{p})) - 1 \leq (1+u)(1+c) - 1$$

$$\mathrm{e}_{\mathrm{rel}}(1-p, 1 \ominus \tilde{p}) \leq (1+u)(1+\mathrm{e}_{\mathrm{rel}}(p,\tilde{p})\frac{p}{1-p}) - 1 \leq (1+u)(1+c) - 1$$

$$\mathrm{e}_{\mathrm{rel}}(n(1-p), n \odot (1 \ominus \tilde{p})) \leq (1+u)^2(1+c) - 1$$

Remark: If $s, t \in \mathbb{N}$ with $t + 2 \leq 2^{s-1}$ and $n \in \{2, \ldots, 2^{t+1}\}$ we have $n \in C_{s,t}$.

Example 3.3. If $t = 52$, $c = 2^{-36}$, $p \leq \frac{2^{16}}{1+2^{16}} \approx 0.999985$, then $\frac{p}{1-p} \leq 2^{16}$ and we get

$$c\frac{p}{1-p} \leq 2^{-20}$$

3.4 Error bounds for the deviance part $\mathrm{bd0}(k, np)$ in case of $|k - np| < 0.1 * |k + np|$ and $\mathrm{e_{rel}}(k, np) \geq c$

Now we estimate the error propagation by the function `bd0`. The function that we want to approximate by `bd0` is the function $\mathbb{N} \times]0, \infty[\ni (k, x) \mapsto k \log(k/x) + x - k$. We will use that for every $k \in \mathbb{N}$, the function $]0, \infty[\ni x \mapsto k \log(k/x) + x - k$ is nonnegative, convex and $= 0$ if $x = k$. In this section, we examine the error propagation in the evaluation of $\mathrm{bd0}(k, np)$ in case of $|k - np| < 0.1 * |k + np|$ and $\mathrm{e_{rel}}(k, np) \geq c$, where $c \in]0, \infty[$. In this case, the following C code fragment is equivalent to the program that R uses.

```
double ej, s, s1, v;
int j;
v = (x-np)/(x+np);
s = (x-np)*v;
if(fabs(s) < DBL_MIN) return s;
ej = 2*x*v;
v = v*v;
for (j = 1; j < 1000; j++) {
    ej *= v;
    s1 = s+ej/((j<<1)+1);
    if (s1 == s) return s1;
    s = s1;
    }
}
```

This program evaluates the following representation of the function $\mathrm{bd0}$, which is valid for $x, y \in \mathbb{R}$ with $|(x - y)/(x + y)| < 1$

$$
\begin{aligned}
\mathrm{bd0}(x, y) &= x \log(x/y) - x + y \\
&= x \log\left(\frac{1 + \frac{x-y}{x+y}}{1 - \frac{x-y}{x+y}}\right) - x + y \\
&= x \left(\log\left(1 + \frac{x-y}{x+y}\right) - \log\left(1 - \frac{x-y}{x+y}\right)\right) - x + y \\
&= 2x \sum_{k=0}^{\infty} \left(\frac{1}{2k+1} (\frac{x-y}{x+y})^{2k+1}\right) - x + y \\
&= 2x \frac{x-y}{x+y} - x + y + 2x \sum_{k=1}^{\infty} \frac{1}{2k+1} (\frac{x-y}{x+y})^{2k+1} \\
&= \frac{(x-y)^2}{x+y} + 2x \sum_{k=1}^{\infty} \frac{1}{2k+1} (\frac{x-y}{x+y})^{2k+1}
\end{aligned}
$$

In the third step the power series expansion $\log(1+x) = \sum_{k=1}^{\infty}(-1)^{k+1}\frac{x^k}{k}$ was used, which is valid for $x \in]-1,1[$.

Our first lemma gives bounds for the error propagation in the initial steps of the algorithm.

Lemma 3.4. *Let $x, y \in]0,\infty[$, $\tilde{x}, \tilde{y} \in]0,\infty[\cap C$, $c_1, c_2 \in [0, 2/(1+u)-1[$, $c_3 \in]0,\infty[$ with $\mathrm{e_{rel}}(x,\tilde{x}) \leq c_1, \mathrm{e_{rel}}(y,\tilde{y}) \leq c_2, \mathrm{e_{rel}}(x,\tilde{y}) \geq c_3$. We further assume $(1-c_2)(1+c_3/(1+c_3)), (1+c_3)(1-c_2/(1-c_2)) > 1$. Let $s := \tilde{x} \oplus \tilde{y}$, $d := \tilde{x} \ominus \tilde{y}$, $v := d \oslash s$, $e_1 := ((2 \odot \tilde{x}) \odot v) \odot (v \odot v)$ and*

$$K_1 := (1 + \max(c_1, c_2))(1+u) - 1$$

$$K_2 := (1+u)(1 + \frac{c_1}{(1-c_2)(1+c_3/(1+c_3)) - 1} + \frac{c_2}{(1+c_3)(1-c_2/(1-c_2)) - 1}) - 1$$

Let $\tilde{x}+\tilde{y}, \tilde{x}-\tilde{y}, d/s, dv, v^2, 2\tilde{x}, (2 \odot \tilde{x})v \in R$. Then

$$\mathrm{e_{rel}}(\frac{(x-y)^2}{x+y}, d \odot v) \leq (1+u)^2(1+K_2)^2/(1-K_1) - 1$$

$$\mathrm{e_{rel}}(\left(\frac{x-y}{x+y}\right)^2, v \odot v) \leq (1+u)^3(1+K_2)^2/(1-K_1)^2 - 1$$

$$\mathrm{e_{rel}}(2x\frac{x-y}{x+y}, (2 \odot \tilde{x}) \odot v) \leq (1+u)^3(1+c_1)(1+K_2)/(1-K_1) - 1$$

and, if $((2 \odot \tilde{x}) \odot v)(v \odot v), e_1/3 \in R$

$$\mathrm{e_{rel}}(2x\left(\frac{x-y}{x+y}\right)^3/3, e_1 \oslash 3) \leq (1+u)^8(1+c_1)(1+K_2)^3/(1-K_1)^3 - 1$$

Proof. By Lemma 2.36, we get

$$\mathrm{e_{rel}}(x+y, s) \leq K_1$$

By Lemma 2.33, we get

$$\mathrm{e_{rel}}(x-y, d) \leq K_2$$

and hence by Lemma 2.35, we get

$$\mathrm{e_{rel}}(\frac{x-y}{x+y}, v) \leq (1+u)(1+K_2)/(1-K_1) - 1$$

From these inequalities with Lemmas 2.34, 2.35 we get the propositions. □

Lemma 3.5. *Let $x, y \in]0,\infty[$, $\tilde{x}, \tilde{y} \in]0,\infty[\cap C$, $c_1, c_2 \in [0, 2/(1+u) - 1[$, $c_3 \in]0,\infty[$ with $\mathrm{e}_{\mathrm{rel}}(x, \tilde{x}) \leq c_1$, $\mathrm{e}_{\mathrm{rel}}(y, \tilde{y}) \leq c_2$, $\mathrm{e}_{\mathrm{rel}}(x, \tilde{y}) \geq c_3$ and $|\tilde{x} \ominus \tilde{y}| < (1 \oslash 10) \odot (\tilde{x} \oplus \tilde{y})$. We define*

$$K_1 := (1+u)(1+\max(c_1, c_2)) - 1$$

$$K_2 := (1+u)(1 + \frac{c_1}{(1-c_2)(1+c_3/(1+c_3)) - 1} + \frac{c_2}{(1+c_3)(1-c_2/(1-c_2)) - 1}) - 1$$

$$K_3 := 12\frac{1-K_2}{1+K_1}$$

$$K_4 := (1+u)^2(1+K_2)^2/(1-K_1) - 1$$

$$K_5 := (1+u)^8(1+c_1)(1+K_2)^3/(1-K_1)^3 - 1$$

Let $s := \tilde{x} \oplus \tilde{y}$, $d := \tilde{x} \ominus \tilde{y}$, $v := d \oslash s$, $a := d \odot v$ and $e_1 := ((2 \odot \tilde{x}) \odot v) \odot (v \odot v)$, We assume [$2, 3, 8, 10 \in C$ and] $\tilde{x}+\tilde{y}, \tilde{x}-\tilde{y}, d/s, dv, v^2, 2\tilde{x}, (2\odot\tilde{x})v, ((2\odot\tilde{x})\odot v)(v\odot v), e_1/3, a+(e_1 \oslash 3) \in R$ and $(1 \oslash 8) \odot (\tilde{x} \oplus \tilde{y}) = (\tilde{x} \oplus \tilde{y})/8$. We further assume $(1-c_2)(1+c_3/(1+c_3)), (1+c_3)(1-c_2/(1-c_2)) > 1$ and $K_1, K_2 < 1, K_3 > 1$. Then we have

$$\mathrm{e}_{\mathrm{rel}}\left(\frac{(x-y)^2}{x+y} + 2x\left(\frac{x-y}{x+y}\right)^3/3, a \oplus (e_1 \oslash 3)\right) \leq (1+u)(1 + \frac{K_4}{1-K_3^{-1}} + \frac{K_5}{K_3 - 1}) - 1$$

If $c_1 = 0, c_2 = 2^{-25}, c_3 = 2^{-18}$, Mathematica yields

$$(1+u)\left(\frac{K_4}{1-K_3^{-1}} + \frac{K_5}{K_3 - 1}\right) + u < 1.95 \cdot 10^{-2}$$

Proof. From Lemma 2.36 we have $\mathrm{e}_{\mathrm{rel}}(x+y, \tilde{x} \oplus \tilde{y}) \leq K_1$ and from Lemma 2.33 we have $x \neq y$ and $\mathrm{e}_{\mathrm{rel}}(x-y, \tilde{x} \ominus \tilde{y}) \leq K_2$. From Lemma 2.29 we get

$$|x-y| \leq |\tilde{x} \ominus \tilde{y}|/(1-K_2)$$

$$x+y \geq (\tilde{x} \oplus \tilde{y})/(1+K_1)$$

We further have

$$|\tilde{x} \ominus \tilde{y}| < (1 \oslash 10) \odot (\tilde{x} \oplus \tilde{y}) < (1 \oslash 8) \odot (\tilde{x} \oplus \tilde{y}) = (\tilde{x} \oplus \tilde{y})/8$$

and therefore $\frac{|\tilde{x} \ominus \tilde{y}|}{\tilde{x} \oplus \tilde{y}} < 1/8$. We therefore have

$$\frac{|x-y|}{x+y} \leq \frac{|\tilde{x} \ominus \tilde{y}|(1+K_1)}{(1-K_2)(\tilde{x} \oplus \tilde{y})} \leq \frac{1+K_1}{8(1-K_2)}$$

and hence

$$\left|\frac{\frac{(x-y)^2}{x+y}}{2x\left(\frac{x-y}{x+y}\right)^3/3}\right| = \frac{3(x+y)^2}{2x|x-y|} \geq \frac{3(x+y)}{2|x-y|} \geq 12\frac{1-K_2}{1+K_1} = K_3$$

From Lemma 3.4 we have

$$\mathrm{e_{rel}}(\frac{(x-y)^2}{x+y}, s) \leq K_4$$

$$\mathrm{e_{rel}}(2x\left(\frac{x-y}{x+y}\right)^3/3, e_1 \oslash 3) \leq K_5$$

Therefore, by Lemma 2.38 we get the proposition. □

Lemma 3.6. *Let $s, t \in \mathbb{N}$ with $4t + 8 \leq 2^{s-1}$. Let $\oplus = \oplus_{s,t}$, $\odot = \odot_{s,t}$, $\ominus = \ominus_{s,t}$, $\oslash = \oslash_{s,t}$. Let $n \in \mathbb{N}$ with $\{1, \ldots, 2n+1\} \subseteq C_{s,t}$, i.e. with $2n + 1 \leq 2^{t+1}$, $\tilde{x}, \tilde{y} \in]0, \infty[\cap C_{s,t}$ with $\tilde{x} \neq \tilde{y}$. We assume $\tilde{x} \geq 1, \tilde{y} \geq 1/2$ and $\tilde{x}, \tilde{y} \leq 2^{t+1}$ and $|\tilde{x} \ominus \tilde{y}| < 2^{-3} \odot (\tilde{x} \oplus \tilde{y})$. We define $s := \tilde{x} \oplus \tilde{y}$, $d := \tilde{x} \ominus \tilde{y}$, $v := d \oslash s$. Let $s_0 := d \odot v, e_0 := (2 \odot \tilde{x}) \odot v$ and $e_j := e_{j-1} \odot (v \odot v), z_j := e_j \oslash (2j+1), s_j := s_{j-1} \oplus z_j$ for $j \in \{1, \ldots, n\}$.*

Then for $k \in \{1, \ldots, n\}$ we have $s_{k-1} + z_k \in R_{s,t}$ and if $6k \geq t + 11$ also $s_k = s_{k-1}$.

Proof. Because of $\tilde{x} \neq \tilde{y}$ we have $|d| \geq \varepsilon_t/2 = 2^{-t-1}$. Furthermore because of $|s| \leq 2^{t+2}$ we have $2^{-2t-3} \leq |v| \leq 2^{-3}$. To prove the proposition we first consider the case of $d < 0$. In this case we use the inequality

$$e_0 = (2 \odot \tilde{x}) \odot (d \oslash s) \geq (2 \odot \tilde{x}) \odot (d \oslash \tilde{x}) \geq 2d(1+u_t)^2 \geq 2d(1+2^{-3})^2 \geq 3d$$

which is valid if $t \geq 2$, and the inequality $e_0 \geq (2 \odot \tilde{x}) \odot (d \oslash \tilde{x}) = 2d \geq 3d$ if $t = 1$. Let $m_1, m_2 \in \mathbb{Z}$ with $-2^{m_1+1} \leq d < -2^{m_1}$ and $-2^{m_2+1} \leq v < -2^{m_2}$. Then we get $e_0 \geq -3 \cdot 2^{m_1+1}$ and $v \odot v \leq 2^{2(m_2+1)}$ and thus

$$e_1 \geq -3 \cdot 2^{m_1+1+2(m_2+1)}$$

$$z_1 = e_1 \oslash 3 \geq e_1/2 \geq -3 \cdot 2^{m_1+2(m_2+1)}$$

and by induction

$$e_k \geq -3 \cdot 2^{m_1+1+2k(m_2+1)}$$

$$z_k = e_k \oslash (2k+1) \geq 2^{-2} e_k \geq -2^{m_1+1+2k(m_2+1)}$$

for $k \in \{2, \ldots, n\}$. Furthermore we have

$$s_0 = d \odot v \geq 2^{m_1+m_2}$$

Because of $|v| \leq 2^{-3}$ we have $m_2 \leq -4$. Hence we get the inequalities

$$\begin{aligned} s_0 + z_1, s_1 &\geq (2^{m_1+m_2} - 3 \cdot 2^{m_1+2(m_2+1)})(1-u_t) \\ &\geq (2^{m_1+m_2} - 3 \cdot 2^{m_1+2(m_2+1)})/2 \\ &= 2^{m_1+m_2-1}(1 - 3 \cdot 2^{2+m_2}) \\ &\geq 2^{m_1+m_2-1}(1 - 3 \cdot 2^{-2}) \\ &= 2^{m_1+m_2-3} \end{aligned}$$

and with Lemma 2.27

$$\begin{aligned} s_{k-1} + z_k &\geq 2^{m_1+m_2-3} - \sum_{j=2}^{\infty} 2^{m_1+2+2j(m_2+1)} \\ &= 2^{m_1+m_2-3} - 2^{m_1+2+4(m_2+1)} \sum_{j=0}^{\infty} 2^{2j(m_2+1)} \\ &= 2^{m_1+m_2-3} - 2^{m_1+2+4(m_2+1)}/(1 - 2^{2(m_2+1)}) \\ &= 2^{m_1+m_2-3}(1 - 2^{9+3m_2}/(1 - 2^{2(m_2+1)})) \\ &\geq 2^{m_1+m_2-3}(1 - 2^{-3}/(1 - 2^{-6})) \\ &\geq 2^{m_1+m_2-4} \end{aligned}$$

for $k \in \{2, \ldots, n\}$. Because of $|v| \geq 2^{-2t-3}$ and $|d| \geq 2^{-t-1}$ we have $m_1 \geq -t-1$ and $m_2 \geq -2t-3$. Hence we get $2^{m_1+m_2-4} \geq 2^{-3t-8} \geq 2^{2-2^{s-1}}$. Because of $s_{k-1} \geq 2^{m_1+m_2-4}$ and $|z_k| \leq 2^{m_1+1+2k(m_2+1)}$ we get $s_{k-1} \oplus z_k = s_{k-1}$ if $6k \geq t+11$. Obviously, in case of $d < 0$ we also have the inequality $s_k \leq s_0$ and hence $s_{k-1} + z_k \in R_{s,t}$ for every $k \in \{1, \ldots, n\}$.

Now we consider the case $d > 0$ where we have $s_k \geq s_0$ for $k \in \{1, \ldots, n\}$. Because of $s_0 \leq 2^{-3}d \leq 2^{-6}s \leq 2^{t-4}$, $e_0 \leq 2^{t-1}$ and $v \odot v \leq 2^{-6}$ we also get $z_k \leq 2^{t-2-6k}$ and hence with Lemma 2.26 we get $s_k \leq 2^{t-3}$ and $s_{k-1} + z_k \leq 2^{t-2} \leq 2^{2^{s-1}-1}$ for $k \in \{1, \ldots, n\}$. Let $m_1, m_2 \in \mathbb{Z}$ with $2^{m_1} \leq d \leq 2^{m_1+1}$ and $2^{m_2} \leq v \leq 2^{m_2+1}$. Then we have $s_{k-1} \geq s_0 = d \odot v \geq 2^{m_1+m_2}$ and $e_0 = (2 \odot \tilde{x}) \odot (d \oslash s) \leq (2 \odot \tilde{x}) \odot (d \oslash \tilde{x}) \leq 2d(1+u_t)^2 \leq 4d \leq 2^{m_1+2}$ and $z_k \leq 2^{m_1+1+2k(m_2+1)}$. From this, if $6k \geq t+7$ we get $s_{k-1} \oplus z_k = s_{k-1}$. □

Now we are able to prove the main result about the deviance part $\mathrm{bd0}(k, np)$ in case of $|x - np| < 0.1 * (k + np)$ and $\mathrm{e_{rel}}(k, \tilde{x}) \geq c$.

Theorem 3.7. *Let $s, t \in \mathbb{N}$ with $4t + 8 \leq 2^{s-1}$. Let $\oplus = \oplus_{s,t}$, $\odot = \odot_{s,t}$, $\ominus = \ominus_{s,t}$, $\oslash = \oslash_{s,t}$. Let $n \in \mathbb{N}$ with $\{1, \ldots, 2n+1\} \subseteq C_{s,t}$, i.e. with $2n+1 \leq 2^{t+1}$, i.e. with $n < 2^t$, $y, x \in]0, \infty[$ and $\tilde{y}, \tilde{x} \in]0, \infty[\cap C_{s,t}$, $c_1, c_2 \in [0, 1/2]$, $c_3 \in]0, \infty[$ with $\tilde{x} \neq \tilde{y}$ and $\mathrm{e_{rel}}(x, \tilde{x}) \leq c_1, \mathrm{e_{rel}}(y, \tilde{y}) \leq c_2, \mathrm{e_{rel}}(x, \tilde{y}) \geq c_3$. We assume $x, \tilde{x} \geq 1$ and $y, x, \tilde{y}, \tilde{x} \leq 2^{t+1}$ and $|\tilde{x} \ominus \tilde{y}| < 2^{-3} \odot (\tilde{x} \oplus \tilde{y})$.*

We define $s := \tilde{x} \oplus \tilde{y}$, $d := \tilde{x} \ominus \tilde{y}$, $v := d \oslash s$. *Let* $s_0 := d \odot v, e_0 := (2 \odot \tilde{x}) \odot v$ *and* $e_j := e_{j-1} \odot (v \odot v), z_j := e_j \oslash (2j+1), s_j := s_{j-1} \oplus z_j$ *for* $j \in \{1, \ldots, n\}$. *We further assume* $(1-c_2)(1+c_3/(1+c_3)), (1+c_3)(1-c_2/(1-c_2) > 1$ *and* $(1+c_1)(1+K_3)^{2m+3}(1+u_t)^{2m+5}-1 < 1$. *Let*

$$K_1 := (1 + \max(c_1, c_2))(1 + u_t) - 1$$

$$K_2 := (1+u)(1 + \frac{c_1}{(1-c_2)(1+c_3/(1+c_3)) - 1} + \frac{c_2}{(1+c_3)(1-c_2/(1-c_2)) - 1}) - 1$$

$$K_3 := \frac{1+K_2}{1-K_1}(1+u_t) - 1$$

Let $\delta = 2^{8-2^{s-1}+4t+r_1+r_2}$ *where* $r_1, r_2 \in \mathbb{N}$ *with* $2n+1 \leq 2^{r_1}$ *and* $r_1 \leq 2t + 2^{s-1} - 10$ *and* $1/(2-(1+c_1)(1+K_3)^{2n+1}(1+u_t)^{2n+3}) \leq 2^{r_2}$. *Then*

$$\begin{aligned} & \left| \frac{(x-y)^2}{x+y} + 2x \sum_{j=1}^{n} (\frac{x-y}{x+y})^{2j+1}/(2j+1) - s_n \right| \\ \leq & \left(\frac{(x-y)^2}{x+y} + 2x \sum_{j=1}^{n} \left| \frac{x-y}{x+y} \right|^{2j+1} /(2j+1) \right) ((1+c_1)(1+K_3)^{2n+1}(1+u_t)^{3n+3} - 1) \\ & +2(n+1)\delta(1+u_t)^n \end{aligned}$$

Furthermore we have $s_j = s_{j-1}$ *for every* $j \in \{1, \ldots, n\}$ *with* $6j \geq t + 11$.

Proof. Because of $4t + 8 \leq 2^{s-1}$ we have $s \geq 5$ and therefore $2^{-3} \odot z = 2^{-3}z$ for every $z \in \mathrm{F}_{s,t}$ with $z \geq 1$ and $z \ominus 3/4 = z - 3/4$ for every $z \in \mathrm{F}_{s,t}$ with $2^{t-1} \geq z \geq 1$. From $|\tilde{x} \ominus \tilde{y}| < 2^{-3} \odot (\tilde{x} \oplus \tilde{y})$ and $\tilde{x} \geq 1$ we get $\tilde{y} > 3/4$ because otherwise would

$$|\tilde{x} \ominus \tilde{y}| \geq \tilde{x} \ominus \frac{3}{4} \geq \tilde{x} \ominus \frac{\tilde{x}}{2} = \tilde{x}/2 \geq \tilde{x}/4 = 2^{-3} \odot (\tilde{x} \oplus \tilde{x}) \geq 2^{-3} \odot (\tilde{x} \oplus \tilde{y})$$

if $\tilde{x} \geq 3/2$ and

$$|\tilde{x} \ominus \tilde{y}| \geq \tilde{x} \ominus \frac{3}{4} = \tilde{x} - \frac{3}{4} \geq \tilde{x}/4 \geq 2^{-3} \odot (\tilde{x} \oplus \tilde{y})$$

if $\tilde{x} < 3/2$.

Hence with $c_2 \leq 1/2$ we get $y \geq 1/2$. We have $2, 3 \in C_{s,t}$, $1 \leq \tilde{x} + \tilde{y}, \tilde{x} \oplus \tilde{y} \leq 2^{t+2}$, $2^{-t-1} = \varepsilon_t/2 \leq |\tilde{x} - \tilde{y}| \leq 2^{t+2}$, $2^{-t-1} \leq |\tilde{x} \ominus \tilde{y}| < 2^{-3}(\tilde{x} \oplus \tilde{y}) \leq 2^{t-1}$, $2^{-2t-3} \leq |d/s|, |v| \leq 2^{-3}$, $2^{-3t-4} \leq dv \leq 2^{t-4}$,$2^{-4t-6} \leq v^2 \leq 2^{-6}$, $2 \leq 2\tilde{x} = 2 \odot \tilde{x} \leq 2^{t+2}$, $2^{-2t-2} \leq |(2 \odot \tilde{x})v| \leq 2^{t-1}$. The proof of these inequalities is mostly very easy. For example the inequality $|v| \leq 2^{-3}$ is valid because $-2^{-3} \leq d \leq 2^{-3}s$ and hence $-2^{-3} = (-2^{-3}s) \oslash s \leq v \leq (2^{-3}s) \oslash s = 2^{-3}$. From

all these inequalities we get $\tilde{x}+\tilde{y}, \tilde{x}-\tilde{y}, d/s, dv, v^2, 2\tilde{x}, (2 \odot \tilde{x})v \in \pm[2^{2-2^{s-1}}, 2^{2^{s-1}-1}] \subseteq R_{s,t}$. Hence we can use Lemma 3.4 and get

$$\mathrm{e_{rel}}(\frac{(x-y)^2}{x+y}, d \odot v) \leq (1+K_2)(1+K_3)(1+u_t)-1$$

$$\mathrm{e_{rel}}(\left(\frac{x-y}{x+y}\right)^2, v \odot v) \leq (1+K_3)^2(1+u_t)-1$$

$$\mathrm{e_{rel}}(2x\frac{x-y}{x+y}, (2 \odot \tilde{x}) \odot v) \leq (1+c_1)(1+K_3)(1+u_t)^2-1$$

Lemma 3.6 yields $s_{k-1}+z_k \in R_{s,t}$ for $k \in \{1,\ldots,n\}$. We want to apply Lemma 2.40 and therefore now want to show that there exists $m \in \{0,\ldots,n\}$ with

$$e_{j-1} \cdot (v \odot v), e_j/(2j+1) \in R_{s,t} \text{ for } j \in \{1,\ldots,m\}$$

and

$$|z_j|, |2x\left(\frac{x-y}{x+y}\right)^{2j+1}/(2j+1)| \leq \delta \text{ for } j \in \{m+1,\ldots,n\}$$

At first, we remark that the values $|e_j|$, $|e_j/(2j+1)|$, $|z_j|$ and $\left|2x(\frac{x-y}{x+y})^{2j+1})/(2j+1)\right|$ are decreasing when j increases. From that we get that all of these values for every $j \in \{1,\ldots,n\}$ are bounded from above by $2^{2^{s-1}-1} \leq \max \mathrm{F}_{s,t}$. Let $\gamma := \delta 2^{-r_2}$. Then $\gamma \geq 2^{2-2^{s-1}} = \min \mathrm{Range_{s,t}} \cap]0,\infty[$ and hence in case of $|e_{j-1}|(v \odot v), |e_j|/(2j+1) \geq \gamma$ for every $j \in \{1,\ldots,n\}$, we are done. Otherwise let $m \in \{0,\ldots,n-1\}$ with $|e_{j-1}|(v \odot v), |e_j|/(2j+1) \geq \gamma$ for $j \in \{1,\ldots,m\}$ and with $|e_m|(v \odot v) < \gamma$ or $|e_{m+1}|/(2m+3) < \gamma$. As $|e_m|(v \odot v) \leq \gamma$ implies $|e_{m+1}| \leq \gamma$, we then have $|e_{m+1}|/(2m+3) < \gamma$ too. Thus $|z_{m+1}| \leq \gamma \leq \delta$. It remains to show that $\left|2x(\frac{x-y}{x+y})^{2m+3})/(2m+3)\right| \leq \delta$. We have $|e_m| \geq \gamma$. If $m \in \{1,\ldots,n-1\}$ we get this from $|e_{m-1}|(v \odot v) \geq \gamma$, if $m=0$ we get this with $r_1 \leq 2t+2^{s-1}-10$ and $|e_0| \geq 2^{-2t-2}$. Hence we have $|e_{m+1}| = |e_m \odot (v \odot v)| \geq \gamma \cdot 2^{-4t-6} = 2^{2-2^{s-1}+r_1}$ and $|e_{m+1}|/(2m+3) \geq 2^{2-2^{s-1}}$. Hence $e_{m+1}/(2m+3) \in R_{s,t}$ and with Lemma 2.36 and Lemma 2.35 we get

$$\begin{aligned}
& \mathrm{e_{rel}}\left(2x(\frac{x-y}{x+y})^{2m+3})/(2m+3), z_{m+1}\right) \\
\leq\ & (1+c_1)(1+K_3)(1+u_t)^2(1+K_3)^{2m+2}(1+u_t)^{m+1}(1+u_t)^{m+2}-1 \\
=\ & (1+c_1)(1+K_3)^{2m+3}(1+u_t)^{2m+5}-1
\end{aligned}$$

From that we get

$$|2x(\frac{x-y}{x+y})^{2m+3})/(2m+3)| \leq |z_{m+1}|/(2-(1+c_1)(1+K_3)^{2m+3}(1+u_t)^{2m+5}) \leq \gamma 2^{r_2} = \delta$$

because of $(1+c_1)(1+K_3)^{2m+3}(1+u_t)^{2m+5}-1<1$.

Thus we are allowed to apply Lemma 2.40 which because of $K_3 \geq K_2$ and hence

$$\begin{aligned}\max\left((1+K_2)(1+K_3)(1+u_t),(1+c_1)(1+K_3)^{2n+1}(1+u_t)^{2n+3}\right)\\ =(1+c_1)(1+K_3)^{2n+1}(1+u_t)^{2n+3}\end{aligned}$$

yields the proposed bound.

□

An Example for the bounds which the last Theorem yields:

Example 3.8. If $c_1=0$, $c_2=2^{-25}$, $c_3=2^{-18}$, Mathematica yields

$$K_2=0.00787, K_3=0.00787, (1+K_3)^{100}-1=1.190918$$

Hence, the relative error bound unfortunately is rather big.

We could improve the bound which the last Theorem yields if we used Lemma 2.40 with $k_0=2$ instead of $k_0=1$ and with $a=\frac{(x-y)^2}{x+y}+2x(\frac{x-y}{x+y}^3/3$ and the error bound for a which is stated in Lemma 3.5.

The following lemma states an upper bound for the number of cycles that the "for" loop in Loader's algorithm takes until it quits.

Lemma 3.9. *Let $s,t\in\mathbb{N}$ with $4t+8\leq 2^{s-1}$. Let $\oplus=\oplus_{s,t}$, $\odot=\odot_{s,t}$, $\ominus=\ominus_{s,t}$, $\oslash=\oslash_{s,t}$. Let $n\in\mathbb{N}$ with $\{1,\ldots,2n+1\}\subseteq C_{s,t}$, i.e. with $2n+1\leq 2^{t+1}$, i.e. with $n<2^t$, $y,x\in\,]0,\infty[$ and $\tilde{y},\tilde{x}\in\,]0,\infty[\cap C_{s,t}$, $c_1,c_2\in[0,1/2]$, $c_3\in\,]0,\infty[$ with $\tilde{x}\neq\tilde{y}$ and $\mathrm{e}_{\mathrm{rel}}(x,\tilde{x})\leq c_1, \mathrm{e}_{\mathrm{rel}}(y,\tilde{y})\leq c_2, \mathrm{e}_{\mathrm{rel}}(x,\tilde{y})\geq c_3$. We assume $x,\tilde{x}\geq 1$ and $y,x,\tilde{y},\tilde{x}\leq 2^{t+1}$ and $|\tilde{x}\ominus\tilde{y}|<2^{-3}\odot(\tilde{x}\oplus\tilde{y})$. We define $s:=\tilde{x}\oplus\tilde{y}$, $d:=\tilde{x}\ominus\tilde{y}$, $v:=d\oslash s$. Let $s_0:=d\odot v, e_0:=(2\odot\tilde{x})\odot v$ and $e_j:=e_{j-1}\odot(v\odot v), z_j:=e_j\oslash(2j+1), s_j:=s_{j-1}\oplus z_j$ for $j\in\{1,\ldots,n\}$. Then $s_{j+1}=s_j$ for every $j\in\mathbb{N}$ with $6j\geq 5t+8$.*

For example, if $t=52$, we have $5t+8=268$ and hence $6j>5t+8$ iff $j\geq 45$. If $t=23$, we have $5t+8=123$ and therefore $6j>5t+8$ iff $j\geq 21$.

Proof. For every $j\in\mathbb{N}_0$ we have $s_j\geq 2^{-3t-8}$ and hence $s_j\oplus\alpha=s_j$ for every $\alpha\in C_{s,t}$ with $|\alpha|\leq 2^{-3t-8-(t+2)}=2^{-4t-10}$. We further have $|z_j|\leq 2^{t-2-6j}$ for $j\in\mathbb{N}$ and hence $|z_j|\leq 2^{-4t-10}$ and hence $s_j=s_{j-1}\oplus z_j=s_{j-1}$ if $6j\geq 5t+8$. □

3.5 Absolute error bounds for the deviance part $\mathrm{bd0}(x,np)$ in case of $\mathrm{e}_{\mathrm{rel}}(x,np)\leq c$

If $\mathrm{e}_{\mathrm{rel}}(x,np)$ is very small, then the function $bd0$ computes 0 as result because the computations leave the range of the number system. Then, we do not get the bound for the relative error we derived in the last section but have to derive a bound for the absolute error instead.

We derive an estimation for the function $\mathrm{bd0}$, which is based on the monotonicity of the computer-functions.

Theorem 3.10. *Let $s, t \in \mathbb{N}$ with $s \geq 4$ and $j_{\max} \in \mathbb{N}$ with $\{1, \ldots, 2j_{\max}+1\} \subseteq C_{s,t}$. Let $x, y \in]0, \infty[\cap C_{s,t}$ with $x, y \leq 2^{2^{s-1}-2}$ and $|x \ominus y| < 2^{-3} \odot (x \oplus y)$. Let $z := x \oplus y, d := x \ominus y, v := d \oslash z, s_0 := d \odot v, e_0 := (2 \odot x) \odot v$ and $e_j := e_{j-1} \odot (v \odot v), z_j := e_j \oslash (2j+1)$ and $s_j := s_{j-1} \oplus z_j$ for $j \in \{1, \ldots, j_{\max}\}$. Then we have $s_j \geq 0$ for every $j \in \{0, \ldots, j_{\max}\}$.*

Proof. Because of $x, y \leq 2^{2^{s-1}-2}$ we have $|d| \leq 2^{2^{s-1}-2}$. If $d \geq 0$ or $v \odot v = 0$ then obviously $s_j \geq 0$ for every $j \in \{0, \ldots, j_{\max}\}$. Let $d < 0$ and $v \odot v > 0$, and therefore $v < 0$. From Lemma 2.21 we further have $2x \in C_{s,t}$ and therefore $2 \odot x = 2x$. Let $m, n \in \mathbb{Z}$ with $x \in]2^{m-1}, 2^m], y \in]2^{n-1}, 2^m]$. We have $m \leq n$ and $2^m, 2^n \in C_{s,t}$. If we assume $n \geq m+2$ then we get $2^{n-2} \geq 2^m$ and therefore $2^{n-2} \in C_{s,t}$ and we get

$$2^{-3} \odot z \leq 2^{-3} \odot 2^{n+1} = 2^{n-2} = -(2^{n-2} \ominus 2^{n-1}) \leq -d = |d|$$

which is a contradiction to $|d| < 2^{-3} \odot z$. Therefore we have $n \leq m+1$. We now show the inequality $e_0 \geq 2d$. Let $k \in \mathbb{Z}$ with $|d|/(2x) \in]2^{k-1}, 2^k]$. Because of $v \neq 0$ we have $2^{1-2^{s-1}-t} \leq |d|/z \leq |d|/(2x)$. Because of $v \odot v \neq 0$ we further have $2^{1-2^{s-1}-t} \neq |d|/z$. Therefore $k \geq 2 - 2^{s-1} - t$. Because of $0 < |d| < 2^{-3} \odot z$ we have $2^{3-2^{s-1}-t} \leq 2^{-3} \odot z \leq 2^{-3} \odot 2^{n+1}$ and therefore $2^{-3}2^{n+1} > 2^{2-2^{s-1}-t}$ and therefore $|d| < 2^{-3} \odot 2^{n+1} = 2^{n-2}$ and $|d|/(2x) \leq 2^{n-2}/2^m \leq 2^{m-1}/2^m = 2^{-1}$. We get $k \leq -1$ and therefore $-2^k \in C_{s,t}$. Because of $d/(2x) \geq -2^k$ with Lemma 2.3 we get $d \underline{\oslash} (2x) \geq -2^k \geq d/x$ and therefore $2d \leq (2x)(d \underline{\oslash} (2x))$. Because $2d \in C_{s,t}$ we get

$$2d \leq (2x) \underline{\odot} (d \underline{\oslash} (2x)) \leq (2x) \underline{\odot} (d \underline{\oslash} z) \leq e_0$$

Now we show the inequality $2^{-3} \odot z \leq 2^{-2} z$. Let $k_1 \in \mathbb{Z}$ with $2^{-3}z \in]2^{k_1-1}, 2^{k_1}]$. We get $2^{k_1} \in C_{s,t}$ because of $2^{-3}z \leq z \leq 2^{2^{s-1}-1}$ and $2^{-3} \odot z > |d| > 0$ and therefore $2^{-3} \odot z \geq 2^{3-2^{s-1}-t}$ and therefore $2^{-3}z \geq 2^{2-2^{s-1}-t}$. From $2^{-3}z \leq 2^{k_1}$ we get $2^{-3} \overline{\odot} z \leq 2^{k_1} \leq 2^{-2}z$ and therefore $|d| < 2^{-3} \odot z \leq 2^{-2}z$. This implies $|v| \leq 2^{-2}$ and $v \odot v \leq 2^{-4}$.

In case of $|v| \geq 2^{4-2^{s-1}}$ with Lemma 2.21 we get $v \odot v \leq v \odot (-2^{-2}) = 2^{-2}|v|$, while in case of $|v| \leq 2^{4-2^{s-1}}$ because of $s \geq 4$ we get $|v| \leq 2^{-4}$ and therefore $v \odot v \leq 2^{-2}|v|$. We get $2d(v \odot v) \geq -2^{-1}dv$ and therefore $e_1 \geq (2d) \odot (v \odot v) \geq -(d \odot v) = -s_0$.

Let $r := s_0/2$. In case of $s_0 \geq 2^{3-2^{s-1}}$ with Lemma 2.21 we get $s_0 \oslash 2 = r \in C_{s,t}$ and therefore $s_1 \geq s_0 \oplus (-r) = r$. In case of $s_0 \leq 2^{3-2^{s-1}}$ we get $s_0 \oplus z_1 = s_0 + z_1 \geq r$.

With $v \odot v \leq 2^{-4}$ we get $e_j \geq -2^{-3(j-1)}e_1$ for $j \in \{1, \ldots, j_{\max}\}$. We further get

$$s_2 \geq r \oplus (-2^{-3}r) \geq (r - 2^{-3}r)/2 \geq r/4$$

and if $j \in \{2, \ldots, j_{\max}\}$ with $s_j \geq 2^{-2(j-1)}r$ we get

$$s_{j+1} \geq (s_j + z_{j+1})/2 \geq (2^{-2(j-1)} - 2^{-3(j-1)})r/2 \geq 2^{-2j}r$$

Therefore we inductively get $s_j \geq 2^{-2(j-1)}r$ for $j \in \{2, \ldots, j_{\max}\}$ and therefore the proposition. □

Theorem 3.11. *Let $s, t \in \mathbb{N}, n \in \{1, \ldots, t\}$ with $3n > t + 5$ and $j_{\max} \in \mathbb{N}$ with $\{1, \ldots, 2j_{\max} + 1\} \subseteq C_{s,t}$. Let $\ell \in \{1, \ldots, n\}$ and $x \in [2^{\ell-1}, 2^{\ell}] \cap C_{s,t}$. Let $y \in [x - 2^{-n+\ell}, x + 2^{-n+\ell}] \cap C_{s,t}$. Then with $v := (x \ominus y) \oslash (x \oplus y)$, $s_0 := (x \ominus y) \odot v$, $e_0 := (2 \odot x) \odot v$ and $e_j := e_{j-1} \odot (v \odot v)$ and $s_j := s_{j-1} \oplus (e_j \oslash (2j + 1))$ for $j \in \{1, \ldots, j_{\max}\}$ we have*

$$s_j \leq 2^{-2n+\ell+1} \oplus 2^{-3n+\ell+1}$$

for every $j \in \{0, \ldots, j_{\max}\}$.

Proof. We have $x \oplus y \geq x \geq 2^{\ell-1}$.

In case of $y \geq x$ we have $x \ominus y \leq 0$, hence $v \leq 0$ and hence $e_j \leq 0$ and $s_j \leq s_0$ for every $j \in \{0, \ldots, j_{\max}\}$. As $0 \geq x \ominus y \geq -2^{-n+\ell}$ and $x \oplus y \geq 2^{\ell-1}$, we get $v \geq -2^{-n+\ell-1} \oslash 2^{\ell-1} = -2^{-n+1}$, therefore $s_0 = (x \ominus y) \odot v \leq 2^{-2n+\ell+1}$ and hence $s_j \leq 2^{-2n+\ell+1}$ for every $j \in \{0, \ldots, j_{\max}\}$.

In case of $y \leq x$ we have $0 \leq x \ominus y \leq x \ominus (x - 2^{-n+\ell}) = 2^{-n+\ell}$ and hence $0 \leq v \leq 2^{-n+\ell} \oslash 2^{\ell-1} = 2^{-n+1}$. From that we get $e_0 \leq 2^{-n+\ell+2}$. From that and $v \odot v \leq 2^{-2n+2}$ we get $e_j \leq 2^{(-2n+2)j-n+\ell+2}$ and hence $e_j \oslash (2j + 1) \leq 2^{(-2n+2)j-n+\ell+1}$ for every $j \in \{0, \ldots, j_{\max}\}$. For $j \in \{2, \ldots, j_{\max}\}$ we have $(2^{-2n+\ell+1} \oplus 2^{-3n+\ell+3}) \oplus 2^{(-2n+2)j-n+\ell+1} = 2^{-2n+\ell} \oplus 2^{-3n+\ell}$ because of $2^{-2n+\ell+1} \oplus 2^{-3n+\ell+3} \geq 2^{-2n+\ell+1}$ and $2^{(-2n+2)j-n+\ell+1} \leq 2^{-5n+\ell+5} < 2^{-2n+\ell-t}$. As $s_0 = (x \ominus y) \odot v \leq 2^{-2n+\ell+1}$ we get the proposition.

□

Lemma 3.12. *Let $K = \mathbb{R}$, $x, y \in]0, \infty[$, $c \in [0, 1[$ with $\mathrm{e_{rel}}(y, x) \leq c$, which means $y \in]x/(1 + c), x/(1 - c)[$. Then we have $x \log(x/y) - x + y \geq 0$ and*

$$x \log(x/y) - x + y \quad \leq \quad \max\{x \log(1 - c) - x + x/(1 - c), x \log(1 + c) - x + x/(1 + c)\}$$

Proof. For fixed x the function $f : [x/(1 + c), x/(1 - c)] \to \mathbb{R}$ defined by $f(y) := x \log(x/y) - x + y$ has derivatives $f'(y) = 1 - x/y$ and $f''(y) = x/y^2$. Therefore f has a local minimum at $y = x$ with $f(y) = 0$ for $y = x$. Further the function f takes its maximal values at the boundary of its domain $[x/(1 + c), x/(1 - c)]$. □

3.6 Error bounds for the deviance part $\mathrm{bd0}(k, np)$ in case of $|k - np| \geq 0.1 * |k + np|$

In this section we examine the error propagation in the evaluation of $\mathrm{bd0}(k, np)$ in case of $|k - np| \geq 0.1 * |k + np|$. In this case the function evaluates the formula `x*log(x/np)+np-x`, using an approximation `log` of the logarithm.

In the rest of this section let $f :]0, \infty[\to K$ with $f(x \cdot y) = f(x) + f(y)$ for $x, y \in]0, \infty[$ and $\tilde{f} : C \cap]0, \infty[\to C$, $v \in [0, \infty[$ with $\tilde{f}(x) \in K$ and $\mathrm{e_{rel}}(f(x), \tilde{f}(x)) \leq v$ for every $x \in C \cap]0, \infty[$ with $f(x) \in R$.

Theorem 3.13. *Let $x \in]0,\infty[$, $\tilde{x} \in C \cap]0,\infty[$ and $c \in [0,1[$ with $f(\tilde{x}) \in R$ and $\mathrm{e}_{\mathrm{rel}}(x,\tilde{x}) \leq c$. Let $M \in [0,\infty[$ with $|f(1-\varepsilon)| \leq M$ for every $\varepsilon \in [-c,c]$. Then*

$$\mathrm{e}_{\mathrm{rel}}(f(x),\tilde{f}(\tilde{x})) \leq v + \frac{(1+v)M}{|f(x)|}$$

Proof. Let $\varepsilon := \frac{x-\tilde{x}}{x}$. We have

$$|f(\tilde{x})| = |f(x(1-\varepsilon))| \leq |f(x)| + |f(1-\varepsilon)| \leq |f(x)| + M$$

and hence

$$\begin{aligned} |f(x) - \tilde{f}(\tilde{x})| &= |f(\tilde{x}/(1-\varepsilon)) - \tilde{f}(\tilde{x})| \\ &\leq |f(\tilde{x}) - \tilde{f}(\tilde{x})| + |f(1-\varepsilon)| \\ &\leq v|f(\tilde{x})| + M \\ &\leq v|f(x)| + (1+v)M \end{aligned}$$

□

Theorem 3.14. *Let $y \in]0,\infty[$, $x,\tilde{y},\alpha \in C \cap]0,\infty[$ with $x/\tilde{y}, x-\tilde{y}, x+\tilde{y}, f(x \oslash \tilde{y}), \alpha(x \oplus \tilde{y}) \in R$ and $c \in [0,1[$ with $\mathrm{e}_{\mathrm{rel}}(y,\tilde{y}) \leq c$ and $(1+u)/(1-c) < 2$. Let $M_1, M_2, M_2 \in [0,\infty[$ with $|f(t)| \leq M_1$ for every $t \in]0,\infty[$ with $|1-t| \leq (1+u)/(1-c) - 1$ and $|f(t)| \leq M_2$ for every $t \in]0,\infty[$ with $|1-t| \leq c$ and $|f(t)| \geq M_3$ for every $t \in]0,\infty[$ with $|1-t| \geq \alpha\frac{(1-u)^2}{1+u}$. Let $|x \ominus \tilde{y}| \geq \alpha \odot (x \oplus \tilde{y})$ and $M_3 > M_2$. Then*

$$\mathrm{e}_{\mathrm{rel}}\left(f(x/y), \tilde{f}(x \oslash \tilde{y})\right) \leq v + \frac{(1+v)M_1}{M_3 - M_2}$$

In particular, if $K = \mathbb{R}$ with usual order $\leq$ and $f = \log$ we get

$$\mathrm{e}_{\mathrm{rel}}\left(\log(x/y), \tilde{f}(x \oslash \tilde{y})\right) \leq v + (1+v)\frac{\left|\log\left(2 - \frac{1+u}{1-c}\right)\right|}{\log\left(1 + \alpha\frac{(1-u)^2}{1+u}\right) - |\log(1-c)|}$$

if $\log\left(1 + \alpha\frac{(1-u)^2}{1+u}\right) > |\log(1-c)|$.

Proof. From 2.35 we get $\mathrm{e}_{\mathrm{rel}}(x/y, x \oslash \tilde{y}) \leq (1+u)/(1-c) - 1$. With that, from 3.13 we get

$$\mathrm{e}_{\mathrm{rel}}\left(f(x/y), \tilde{f}(x \oslash \tilde{y})\right) \leq v + \frac{(1+v)M_1}{|f(x/y)|}$$

With $\eta := \frac{x-\tilde{y}-(x\ominus\tilde{y})}{x-\tilde{y}}$ we get $|x \ominus \tilde{y}| = |(x-\tilde{y})(1-\eta)| \leq |x-\tilde{y}|(1+u)$ and hence

$$\left|1 - \frac{x}{\tilde{y}}\right| = \frac{|x-\tilde{y}|}{\tilde{y}} \geq \frac{\alpha \odot (x \oplus \tilde{y})}{\tilde{y}(1+u)} \geq \frac{\alpha(x \oplus \tilde{y})(1-u)}{\tilde{y}(1+u)} \geq \alpha\left(1 + \frac{x}{\tilde{y}}\right)\frac{(1-u)^2}{1+u} \geq \alpha\frac{(1-u)^2}{1+u}$$

We get $f(x/\tilde{y}) \geq M_3$. Let $\varepsilon := \frac{y-\tilde{y}}{y}$. We have

$$|f(x/y)| = |f(x/\tilde{y}) + f(1-\varepsilon)| \geq |f(x/\tilde{y})| - |f(1-\varepsilon)| \geq M_3 - M_2$$

Therefore we get

$$\mathrm{e}_{\mathrm{rel}}\left(f(x/y), \tilde{f}(x \oslash \tilde{y})\right) \leq v + \frac{(1+v)M_1}{M_3 - M_2}$$

The inequality for $K = \mathbb{R}, f = \log$ follows from

$$|\log(t)| \leq |\log(1-s)| \text{ for } s \in [0,1[, t \in]0,\infty[\text{ with } |1-t| \leq s$$
$$|\log(t)| \geq \log(1+s) \text{ for } s,t \in]0,\infty[\text{ with } |1-t| \geq s$$

□

3.7 Approximative evaluation of Stirling's Series

In Appendix C we described approximations of the function $\mu :]0,\infty[\to \mathbb{R}$

$$\mu(x) = \log\left(\frac{\Gamma(x+1)}{\left(\frac{x}{\mathrm{e}}\right)^x \sqrt{2\pi x}}\right)$$

by Stirling's Series. Loader's algorithm for the binomial density utilizes the function `stirlerr`, which we displayed in Appendix D.3, to compute approximative values for $\mu(x)$. Depending on how large x is, the function `stirlerr` approximatively evaluates one of the following four partial sums of Stirling's Series:

$$\frac{1}{12x} - \frac{1}{360x^3}, \text{ if } x > 500$$

$$\frac{1}{12x} - \frac{1}{360x^3} + \frac{1}{1260x^5}, \text{ if } 80 < x \leq 500$$

$$\frac{1}{12x} - \frac{1}{360x^3} + \frac{1}{1260x^5} - \frac{1}{1680x^7}, \text{ if } 35 < x \leq 80$$

$$\frac{1}{12x} - \frac{1}{360x^3} + \frac{1}{1260x^5} - \frac{1}{1680x^7} + \frac{1}{1188x^9}, \text{ if } 15 < x \leq 35$$

If $x \in \{1,\ldots,15\}$ the function `stirlerr` returns a value which is stored in an internal table. In this section we derive an error bound for the approximation of $\mu(x)$ by the computed value `stirlerr(`x`)` for $x \in \{16,\ldots,n_{\max}\}$ depending on $n_{\max} \in \mathbb{N}$.

In the first lemma of this section we examine the error propagation of an algorithm that alternates division and subtraction, which we will apply to approximately evaluate Stirling's Series.

Lemma 3.15. *Let* $m \in \mathbb{N}, a_1, \ldots, a_m \in [0, \infty[, b_1, \ldots, b_m \in [0, \infty[\cap C$ *and* $c \in [0, \infty[$ *with* $\mathrm{e_{rel}}(a_1, b_1), \ldots, \mathrm{e_{rel}}(a_m, b_m) \leq c$. *Let* $y_1, \ldots, y_m \in]0, \infty[, z_1, \ldots, z_m \in]0, \infty[\cap C, e_1, \ldots, e_m \in [0, 1[$ *with* $\mathrm{e_{rel}}(y_1, z_1) \leq e_1, \ldots, \mathrm{e_{rel}}(y_m, z_m) \leq e_m$. *Let* $d_1 := b_1$, $q_1 := d_1 \oslash z_1$ *and* $d_k := b_k \ominus q_{k-1}$ *and* $q_k := d_k \oslash z_k$ *for* $k \in \{2, \ldots, m\}$. *We assume*

$$b_1/z_1 \in R, b_k - q_{k-1}, d_k/z_k \in R \text{ for } k \in \{2, \ldots, m-1\}, b_m - q_{m-1} \in R \tag{3.6}$$

Then we have

$$\left| \sum_{i=1}^{m} \left((-1)^{m-i} a_i / \prod_{j=i}^{m-1} y_j \right) - d_m \right| \leq \left((1+c)(1+u)^{2m-2} / \prod_{i=1}^{m-1} (1-e_i) - 1 \right) \sum_{i=1}^{m} a_i / \prod_{j=i}^{m-1} y_j \tag{3.7}$$

and, if $d_m/z_m \in R$

$$\left| \sum_{i=1}^{m} \left((-1)^{m-i} a_i / \prod_{j=i}^{m} y_j \right) - q_m \right| \leq \left((1+c)(1+u)^{2m-1} / \prod_{i=1}^{m} (1-e_i) - 1 \right) \sum_{i=1}^{m} a_i / \prod_{j=i}^{m} y_j \tag{3.8}$$

Proof. In case of $m = 1$ the proposition follows from Lemma 2.35. Let $m \geq 2$. Let $\eta_1 := \frac{b_1/z_1 - q_1}{b_1/z_1}, \varepsilon_1 := 0, \eta_k := \frac{d_k/z_k - q_k}{d_k/z_k}$ and $\varepsilon_k := \frac{b_k - q_{k-1} - d_k}{b_k - q_{k-1}}$ for $k \in \{2, \ldots, m\}$. We have

$$q_1 = b_1/z_1(1 - \eta_1)$$

and

$$q_k = (b_k - q_{k-1})/z_k(1 - \eta_k)(1 - \varepsilon_k) \text{ for } k \in \{2, \ldots, m\}$$

By induction we get

$$q_m = \sum_{i=1}^{m} \left((-1)^{m-i} b_i \prod_{j=i}^{m} ((1 - \varepsilon_j)(1 - \eta_j)) / \prod_{j=i}^{m} z_j \right)$$

and

$$d_m = q_m z_m / (1 - \eta_m) = \sum_{i=1}^{m} \left((-1)^{m-i} b_i (1 - \varepsilon_m) \prod_{j=i}^{m-1} ((1 - \varepsilon_j)(1 - \eta_j)) / \prod_{j=i}^{m-1} z_j \right)$$

Hence, with $\gamma_i := \frac{a_i - b_i}{a_i}$ and $\delta_i := \frac{y_i - z_i}{y_i}$ for $i \in \{1, \dots, m\}$ we get

$$\begin{aligned}
& \left| \sum_{i=1}^{m} \left((-1)^{m-i} a_i / \prod_{j=i}^{m} y_j \right) - q_m \right| \\
= & \left| \sum_{i=1}^{m} \left((-1)^{m-i} a_i / \prod_{j=i}^{m} y_j \left(1 - (1-\gamma_i) \prod_{j=i}^{m} (1-\varepsilon_j)(1-\eta_j)/(1-\delta_j) \right) \right) \right| \\
\leq & \sum_{i=1}^{m} a_i / \prod_{j=i}^{m} y_j \left| 1 - (1-\gamma_i) \prod_{j=i}^{m} ((1-\varepsilon_j)(1-\eta_j)/(1-\delta_j)) \right| \\
\leq & \sum_{i=1}^{m} a_i / \prod_{j=i}^{m} y_j \left((1+c)(1+u)^{2m-1} / \prod_{i=1}^{m} (1-e_i) - 1 \right)
\end{aligned}$$

and

$$\begin{aligned}
& \left| \sum_{i=1}^{m} \left((-1)^{m-i} a_i / \prod_{j=i}^{m-1} y_j \right) - d_m \right| \\
= & \left| \sum_{i=1}^{m} \left((-1)^{m-i} a_i / \prod_{j=i}^{m-1} y_j \left(1 - (1-\gamma_i)(1-\varepsilon_m) \prod_{j=i}^{m-1} (1-\varepsilon_j)(1-\eta_j)/(1-\delta_j) \right) \right) \right| \\
\leq & \sum_{i=1}^{m} a_i / \prod_{j=i}^{m-1} y_j \left| 1 - (1-\gamma_i)(1-\varepsilon_m) \prod_{j=i}^{m-1} ((1-\varepsilon_j)(1-\eta_j)/(1-\delta_j)) \right| \\
\leq & \sum_{i=1}^{m} a_i / \prod_{j=i}^{m-1} y_j \left((1+c)(1+u)^{2m-2} / \prod_{i=1}^{m-1} (1-e_i) - 1 \right)
\end{aligned}$$

□

To be able to practically apply the previous lemma, we have to replace the occuring conditions (3.6) and $d_m/z_m \in R$ by formulas which are easily verfiable in the concrete case of $C = C_{s,t}$. This will be done in the next lemma.

Lemma 3.16. *Let $m \in \mathbb{N}$ with $m \geq 2$, $a_1, \dots, a_m \in]0, \infty[$, $b_1, \dots, b_m \in]0, \infty[\cap C$, $c \in [0, 1[$ with $\mathrm{e_{rel}}(a_1, b_1), \dots, \mathrm{e_{rel}}(a_m, b_m) \leq c$. Let $x \in [1, \infty[$ and $z_1, \dots, z_m \in [1, \infty[\cap C$ and $e, f \in [0, 1[$ with $\mathrm{e_{rel}}(x^2, z_1), \dots, \mathrm{e_{rel}}(x^2, z_{m-1}) \leq f, \mathrm{e_{rel}}(x, z_m) \leq e$. Let $d_1 := b_1, q_1 := d_1 \oslash z_1$ and $d_k := b_k \ominus q_{k-1}$ and $q_k := d_k \oslash z_k$ for $k \in \{2, \dots, m\}$. Let $A, B \in]0, \infty[$ with $[A, B] \subseteq R$ and*

$$a_1(1+c) \leq B \tag{3.9}$$

$$a_1(1-c)/(x^2(1+f)) \geq A \tag{3.10}$$

$$a_{k+1}(1-c) - a_k/x^2 - h \geq A \text{ for } k \in \{1, \dots, m-1\} \tag{3.11}$$

(3.12) $$a_{k+1}(1+c)+h \le B \text{ for } k \in \{1,\dots,m-1\}$$

(3.13) $$(a_{k+1}-a_k/x^2-h)/(x^2(1+f)) \ge A \text{ for } k \in \{1,\dots,m-2\}$$

(3.14) $$(a_m - a_{m-1}/x^2 - h)/(x(1+e)) \ge A$$

with $a_0 := 0$ *and* $h := ((1+c)(1+u)^{2m-1}/(1-\max\{e,f\})^m - 1)\sum_{i=1}^m a_i$. *We define* $e_1,\dots,e_{m-1} := f, e_m := e$ *and* $y_1,\dots,y_{m-1} := x^2, y_m := x$. *Then with* $s := \sum_{i=1}^m a_i$ *we have* (3.7) *and* (3.8).

Proof. At first, we inductively show that for $k \in \{1,\dots,m\}$ the following inequalities are valid

(3.15) $$\sum_{j=1}^{k}(-1)^{k-j}\frac{a_j}{x^{2(k-j)}} \begin{cases} \ge a_k - \frac{a_{k-1}}{x^2} \\ \le a_k \end{cases}$$

(3.16) $$\sum_{j=1}^{k}(-1)^{k-j}\frac{a_j}{x^{2(k-j+1)}} \begin{cases} \ge 0 \\ \le \frac{a_k}{x^2} \end{cases}$$

The case $k=1$ is trivial. If the above inequalities hold for $k \in \{1,\dots,m-1\}$, then

$$\sum_{j=1}^{k+1}(-1)^{k+1-j}\frac{a_j}{x^{2(k+1-j)}} = a_{k+1} - \sum_{j=1}^{k}(-1)^{k-j}\frac{a_j}{x^{2(k-j+1)}} \begin{cases} \ge a_{k+1} - \frac{a_k}{x^2} \\ \le a_{k+1} \end{cases}$$

From $a_{k+1}(1-c) - a_k/x^2 - h \ge A$ we get $a_{k+1} - \frac{a_k}{x^2} \ge 0$. Thus

$$\sum_{j=1}^{k+1}(-1)^{k+1-j}\frac{a_j}{x^{2(k+2-j)}} = \left(\sum_{j=1}^{k+1}(-1)^{k+1-j}\frac{a_j}{x^{2(k+1-j)}}\right)/x^2 \begin{cases} \ge (a_{k+1} - \frac{a_k}{x^2})/x^2 \ge 0 \\ \le \frac{a_{k+1}}{x^2} \end{cases}$$

Thus the induction is complete. We define

$$g_k := \left((1+c)(1+u)^{2k-2}/\prod_{i=1}^{k-1}(1-e_i) - 1\right)\sum_{i=1}^{k} a_i$$
$$h_k := \left((1+c)(1+u)^{2k-1}/\prod_{i=1}^{k}(1-e_i) - 1\right)\sum_{i=1}^{k} a_i$$

for $k \in \{1,\dots,m\}$. Then $g_k, h_k \le h$ for $k \in \{1,\dots,m\}$. Now we inductively show that for $k \in \{1,\dots,m\}$ we have

$$b_k - q_{k-1}, d_k/z_k \in R$$

with $q_0 := 0$. The base of the induction $b_1, d_1/z_1 \in R$ is valid because of

$$A \leq a_1(1-c)/(x^2(1+f)) \leq d_1/z_1 \leq b_1 \leq a_1(1+c) \leq B$$

Let $k \in \{1, \ldots, m-1\}$ with $b_1 - q_0, d_1/z_1, \ldots, b_k - q_{k-1}, d_k/z_k \in R$. Then Lemma 3.15 and (3.16) yield

$$q_k \left\{ \begin{array}{l} \geq -h_k \\ \leq a_k/x^2 + h_k \end{array} \right.$$

Hence

$$b_{k+1} - q_k \left\{ \begin{array}{l} \geq b_{k+1} - a_k/x^2 - h_k \geq a_{k+1}(1-c) - a_k/x^2 - h_k \geq A \\ \leq b_{k+1} + h_k \leq a_{k+1}(1+c) + h_k \leq B \end{array} \right.$$

Hence $b_{k+1} - q_k \in R$ and Lemma 3.15 and (3.15) yield

$$d_{k+1} \left\{ \begin{array}{l} \geq a_{k+1} - \frac{a_k}{x^2} - g_{k+1} \\ \leq a_{k+1} + g_{k+1} \leq B \end{array} \right.$$

Hence

$$d_{k+1}/z_{k+1} \left\{ \begin{array}{l} \geq (a_{k+1} - \frac{a_k}{x^2} - g_{k+1})/z_{k+1} \geq A \\ \leq d_{k+1} \leq B \end{array} \right.$$

and therefore $d_{k+1}/z_{k+1} \in R$. Thus, the induction is complete. Now we are allowed to apply Lemma 3.15 which yields the proposition. □

Now we examine the approximative evaluation of Stirling's Series in the number system $C_{s,t}$. We define $\gamma_1 := 1/12, \gamma_2 := 1/360, \gamma_3 := 1/1260, \gamma_4 := 1/1680, \gamma_5 := 1/1188, \gamma_6 := 691/360360$ and $S_n(x) := \sum_{k=1}^{n} (-1)^{k-1} \gamma_k / x^{2k-1}$ and $h_n(x) := \sum_{k=1}^{n} \gamma_k / x^{2k-1}$ for $n \in \{1, \ldots, 5\}$ and $x \in]0, \infty[$. If $s \geq 5$ we have $\gamma_1, \ldots, \gamma_6 \in F_{s,t}$.

Corollary 3.17. *Let $s, t \in \mathbb{N}$ with $s \geq 5, t \geq 15$ and $b_1, \ldots, b_5 \in]0, \infty[\cap C_{s,t}$ with $\mathrm{e_{rel}}(\gamma_1, b_1), \ldots, \mathrm{e_{rel}}(\gamma_5, b_5) \leq u_t$. Let $x \in [2, 2^{2^{s-2}-8}], y \in [1, \infty[\cap C_{s,t}$ and $e \in [0, 2^{-17}[$ with $\mathrm{e_{rel}}(x, y) \leq e$. Let $z := y \odot y$. Then with $f := (1+e)^2(1+u_t) - 1$ and $g := (1+u_t)^{10}/(1-f)^5 - 1$ we have*

$$|S_2(x) - ((b_0 \ominus b_1 \oslash z) \oslash y)| \leq g h_2(x)$$

$$|S_3(x) - ((b_0 \ominus (b_1 \ominus b_2 \oslash z) \oslash z) \oslash y)| \leq g h_3(x)$$

$$|S_4(x) - ((b_0 \ominus (b_1 \ominus (b_2 \ominus b_3 \oslash z) \oslash z) \oslash z) \oslash y)| \leq g h_4(x)$$

$$|S_5(x) - (b_0 \ominus (b_1 \ominus (b_2 \ominus (b_3 \ominus b_4 \oslash z) \oslash z) \oslash z) \oslash z) \oslash y| \leq g h_5(x)$$

Proof. From Lemma 2.34 we get $\mathrm{e_{rel}}(x^2, z) \leq f$. Because of $t \geq 15$ and $e < 2^{-17}$ with Mathematica we get $(\gamma_4 - \gamma_5/4 - g)/(1+f) > 2^{-14}$ and $\gamma_4(1-u_t) - \gamma_5/4 - g > 2^{-14}$ and $\gamma_1 - \gamma_2/4 - g > 2^{-4}$. We have $[A, B] \subseteq R_{s,t}$ with $A = 2^{2-2^{s-1}}$ and $B = 2^{2^{s-1}-1}$. Now we apply Lemma 3.16 four times, the first time with $m = 2$ and $(a_1, a_2) = (\frac{1}{360}, \frac{1}{12})$, the second time with $m = 3$ and $(a_1, a_2, a_3) = (\frac{1}{1260}, \frac{1}{360}, \frac{1}{12})$, and so on. We need to verify conditions (3.9)- (3.14) with $c = u_t$ and $h = g$. We use that $2^{-11} \leq \gamma_1, \ldots, \gamma_5 \leq 1$.

Verification of (3.9):

$$a_1(1+c) \leq 2a_1 \leq 2 \leq B$$

Verification of (3.10):

$$a_1(1-c)/(x^2(1+f)) \geq 2^{-11} \cdot 2^{-1}/(2^{2^{s-1}-16} \cdot 2) = 2^{3-2^{s-1}} \geq A$$

Verification of (3.11):

$$a_{k+1}(1-c) - a_k/x^2 - h \geq \gamma_4(1-2^{-16}) - \gamma_5/4 - h \geq 2^{-14} \geq A$$

Verification of (3.12):

$$a_{k+1}(1+c) + h \leq 2 + 1 \leq B$$

Verification of (3.13):

$$(a_{k+1} - a_k/x^2 - h)/(x^2(1+f)) \geq (\gamma_4 - \gamma_5/4 - h)/(1+f)2^{16-2^{s-1}} \geq A$$

Verification of (3.14)

$$(a_m - a_{m-1}/x^2 - h)/(x(1+e)) \geq (\gamma_1 - \gamma_2/4 - h)/(2^{2^{s-2}-8} \cdot 2) \geq 2^{3-2^{s-2}} \geq A$$

Lemma 3.16 yields the proposed inequalities. □

In the rest of this section let $K = \mathbb{R}$. We now compare the approximative evaluation of Stirling's series to the value $\mu(x)$.

Corollary 3.18. *Let $s, t \in \mathbb{N}$ with $s \geq 5, t \geq 15$. Let $b_1, \ldots, b_5 \in]0, \infty[\cap C_{s,t}$ with $\mathrm{e_{rel}}(\gamma_1, b_1), \ldots, \mathrm{e_{rel}}(\gamma_5, b_5) \leq u_t$. Let $x_{\min}, x \in [2, 2^{2^{s-2}-8}]$ with $x \geq x_{\min}$ and $y \in [1, \infty[\cap C_{s,t}$ and $e \in [0, 2^{-17}[$ with $\mathrm{e_{rel}}(x, y) \leq e$. Let $z := y \odot y$. Then with $g := (1+u_t)^{10}/(2-(1+e)^2(1+u_t))^5 - 1$ and $L := S_2(x_{\min})$ we have*

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus b_1 \oslash z) \oslash y)\right) \leq \left(gh_2(x_{\min}) + \gamma_3 x_{\min}^{-5}\right)/L$$

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus (b_1 \ominus b_2 \oslash z) \oslash z) \oslash y)\right) \leq \left(gh_3(x_{\min}) + \gamma_4 x_{\min}^{-7}\right)/L$$

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus (b_1 \ominus (b_2 \ominus b_3 \oslash z) \oslash z) \oslash z) \oslash y)\right) \leq \left(gh_4(x_{\min}) + \gamma_5 x_{\min}^{-9}\right)/L$$

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus (b_1 \ominus (b_2 \ominus (b_3 \ominus b_4 \oslash z) \oslash z) \oslash z) \oslash z) \oslash y)\right) \leq \left(gh_5(x_{\min}) + \gamma_6 x_{\min}^{-11}\right)/L$$

Proof. We have

$$\begin{aligned}
\mathrm{e}_{\mathrm{rel}}(\mu(x),(b_0\ominus b_1\oslash z)\oslash y) &= \frac{|\mu(x)-(b_0\ominus b_1\oslash z)\oslash y|}{\mu(x)}\\
&\leq \frac{|\mu(x)-S_2(x)|+|S_2(x)-(b_0\ominus b_1\oslash z)\oslash y|}{S_2(x)}\\
&\leq \frac{gh_2(x)+\gamma_3x^{-5}}{S_2(x)}
\end{aligned}$$

and analogously

$$\mathrm{e}_{\mathrm{rel}}\left(\mu(x),(b_0\ominus(b_1\ominus b_2\oslash z)\oslash z)\oslash y)\right) \leq \left(gh_3(x)+\gamma_4x^{-7}\right)/S_2(x)$$

$$\mathrm{e}_{\mathrm{rel}}\left(\mu(x),(b_0\ominus(b_1\ominus(b_2\ominus b_3\oslash z)\oslash z)\oslash z)\oslash y)\right) \leq \left(gh_4(x)+\gamma_5x^{-9}\right)/S_2(x)$$

$$\mathrm{e}_{\mathrm{rel}}\left(\mu(x),(b_0\ominus(b_1\ominus(b_2\ominus(b_3\ominus b_4\oslash z)\oslash z)\oslash z)\oslash z)\oslash y)\right) \leq \left(gh_5(x)+\gamma_6x^{-11}\right)/S_2(x)$$

When x increases, the right sides of these inequalities are decreasing because

$$h_2(x)/S_2(x) = 1+2\gamma_2/(x^3S_2(x))$$

and $x^3S_2(x)$ is increasing. Therefore we get the proposed inequalities. □

Example 3.19. Let $s,t \in \mathbb{N}$ with $s \geq 5, t \geq 15$ and $b_1,\ldots,b_5 \in]0,\infty[\cap C_{s,t}$ with $\mathrm{e}_{\mathrm{rel}}(\gamma_1,b_1),\ldots,\mathrm{e}_{\mathrm{rel}}(\gamma_5,b_5) \leq u_t$. Let $x \in [2,2^{2^{s-2}-8}]\cap C_{s,t}$ and $z := x\odot x$. Then from Corollary 3.18 and verifications with Mathematica we get the following inequalities.

If $(s,t)=(11,52)$:

$$\mathrm{e}_{\mathrm{rel}}\left(\mu(x),(b_0\ominus b_1\oslash z)\oslash x)\right) \leq 2^{-42},\ \text{if } x>500$$

$$\mathrm{e}_{\mathrm{rel}}\left(\mu(x),(b_0\ominus(b_1\ominus b_2\oslash z)\oslash z)\oslash x)\right) \leq 2^{-44},\ \text{if } x>80$$

$$\mathrm{e}_{\mathrm{rel}}\left(\mu(x),(b_0\ominus(b_1\ominus(b_2\ominus b_3\oslash z)\oslash z)\oslash z)\oslash x)\right) \leq 2^{-47},\ \text{if } x>35$$

$$\mathrm{e}_{\mathrm{rel}}\left(\mu(x),(b_0\ominus(b_1\ominus(b_2\ominus(b_3\ominus b_4\oslash z)\oslash z)\oslash z)\oslash z)\oslash x)\right) \leq 2^{-44},\ \text{if } x>15$$

If $(s,t)=(8,23)$:

$$\mathrm{e}_{\mathrm{rel}}\left(\mu(x),(b_0\ominus b_1\oslash z)\oslash x)\right) \leq 2^{-20},\ \text{if } x>500$$

$$\mathrm{e}_{\mathrm{rel}}\left(\mu(x),(b_0\ominus(b_1\ominus b_2\oslash z)\oslash z)\oslash x)\right) \leq 2^{-20},\ \text{if } x>80$$

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus (b_1 \ominus (b_2 \ominus b_3 \oslash z) \oslash z) \oslash z) \oslash x)\right) \leq 2^{-20}, \text{ if } x > 35$$

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus (b_1 \ominus (b_2 \ominus (b_3 \ominus b_4 \oslash z) \oslash z) \oslash z) \oslash z) \oslash x)\right) \leq 2^{-20}, \text{ if } x > 15$$

If $(s,t) = (15, 63)$:

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus b_1 \oslash z) \oslash x)\right) \leq 2^{-42}, \text{ if } x > 500$$

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus (b_1 \ominus b_2 \oslash z) \oslash z) \oslash x)\right) \leq 2^{-45}, \text{ if } x > 80$$

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus (b_1 \ominus (b_2 \ominus b_3 \oslash z) \oslash z) \oslash z) \oslash x)\right) \leq 2^{-47}, \text{ if } x > 35$$

$$\mathrm{e_{rel}}\left(\mu(x), (b_0 \ominus (b_1 \ominus (b_2 \ominus (b_3 \ominus b_4 \oslash z) \oslash z) \oslash z) \oslash z) \oslash x)\right) \leq 2^{-44}, \text{ if } x > 15$$

Corollary 3.20. *Let $s, t \in \mathbb{N}$. Let $x_{\max} \in [1, \infty[$ and $x_1, x_2 \in C_{s,t} \cap [1, x_{\max}], y_1, y_2, y_3 \in C_{s,t}, c \in [0, 1[$ with $\mathrm{e_{rel}}(\mu(x_1), y_1), \mathrm{e_{rel}}(\mu(x_2), y_2), \mathrm{e_{rel}}(\mu(x_1 - x_2), y_3) \leq c$ and $x_1 \geq x_2 + 1$. We assume $S_1(x_{\max}) - S_2(x_{\max} - 1) + c(S_1(1) + S_2(1)) \leq -2^{2-2^{s-1}}$. Then with $q := S_2(x_{\max} - 1)/S_1(x_{\max})$ and $d := 1/(q-1) + 1/(1-q^{-1})$ we have the inequalities*

$$\mathrm{e_{rel}}(\mu(x_1) - \mu(x_2), y_1 \ominus y_2) \leq (1 + u_t)(1 + dc) - 1$$

$$\mathrm{e_{rel}}(\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2), (y_1 \ominus y_2) \ominus y_3) \leq (1 + u_t)^2 (1 + dc) - 1$$

Proof. The condition $y_1 - y_2 \in [-2^{2^{s-1}-1}, -2^{2-2^{s-1}}] \subseteq R_{s,t}$ is fullfilled because of

$$y_1 - y_2 \geq -y_2 \geq -\mu(x_2)(1 + c) \geq -(1 + c) \geq -2 \geq -2^{2^{s-1}-1}$$

and

$$\begin{aligned} y_1 - y_2 &\leq \mu(x_1)(1 + c) - \mu(x_2)(1 - c) \\ &\leq S_1(x_1)(1 + c) - S_2(x_2)(1 - c) \\ &\leq S_1(x_1)(1 + c) - S_2(x_1 - 1)(1 - c) \\ &\leq S_1(x_{\max}) - S_2(x_{\max} - 1) + c(S_1(1) + S_2(1)) \\ &\leq -2^{2-2^{s-1}} \end{aligned}$$

In the second last step we used that the function $]2, \infty[\ni x \mapsto S_1(x) - S_2(x - 1)$ is increasing. We apply Lemma 2.31 and get that

$$\begin{aligned} & \mathrm{e_{rel}}(\mu(x_1) - \mu(x_2), y_1 \ominus y_2) \\ \leq\ & (1 + u_t)\left(1 + c\left(\frac{1}{|1 - \mu(x_2)/\mu(x_1)|} + \frac{1}{|1 - \mu(x_1)/\mu(x_2)|}\right)\right) - 1 \end{aligned}$$

We have $\mu(x_2)/\mu(x_1) \geq S_2(x_2)/S_1(x_1) \geq S_2(x_1-1)/S_1(x_1)$. The function $]2,\infty[\ni x \mapsto S_2(x-1)/S_1(x)$ is decreasing. Hence we get $\mu(x_2)/\mu(x_1) \geq q$ and $\mu(x_1)/\mu(x_2) \leq q^{-1}$. Therefore we get the first of the proposed inequalites. The second we again get from Lemma 2.31 which this time we are allowed to apply because

$$(y_1 \ominus y_2) - y_3 \geq (0 \ominus 1) - 1 = -2 \geq -2^{2^{s-1}-1}$$

$$(y_1 \ominus y_2) - y_3 \leq y_1 \ominus y_2 \leq -2^{2-2^{s-1}}$$

and hence $(y_1 \ominus y_1) - y_3 \in R_{s,t}$. □

Remark. As the proof shows, in Corollary 3.20 we could also use the weaker precondition

$$S_1(x_{\max})(1+c) - S_2(x_{\max}-1)(1-c) \leq -2^{2-2^{s-1}}$$

instead of

$$S_1(x_{\max}) - S_2(x_{\max}-1) + c(S_1(1)+S_2(1)) \leq -2^{2-2^{s-1}}$$

In order to do that, we had to derive monotonicity of the function $S_1(x_1)(1+c)-S_2(x_1-1)(1-c)$ on an interval depending on c. Using the weaker precondition would allow us to increase $x_{\max}$ given c and s.

Example 3.21. Let $s = 11, t = 52, c = 2^{-42}, x_{\max} = 2^{20}$. Then with Mathematica we verify $S_1(x_{\max}) - S_2(x_{\max}-1) + c(S_1(1)+S_2(1)) \leq -2^{2-2^{s-1}}$ and therefore from Corollary 3.20 we get

$$\mathrm{e}_{\mathrm{rel}}(\mu(x_1) - \mu(x_2), y_1 \ominus y_2) \leq 2^{-21}$$

for $x_1, x_2 \in C_{s,t} \cap [1, 2^{20}[$.

Let $s = 15, t = 63, c = 2^{-42}, x_{\max} = 2^{21}$. The range condition is not fulfilled.

Let $s = 15, t = 63, c = 2^{-42}, x_{\max} = 2^{20}$. We get

$$\mathrm{e}_{\mathrm{rel}}(\mu(x_1) - \mu(x_2), y_1 \ominus y_2) \leq 2^{-21}$$

for $x_1, x_2 \in C_{s,t} \cap [1, 2^{20}[$.

3.8 Computation of the value "lc" in Loader's algorithm

In this section we examine the follwowing function `lc`.

```
double lc(double x, double n, double p){
 double q = 1-p;
 return stirlerr(n)-stirlerr(x)-stirlerr(n-x)-bd0(x,n*p)-bd0(n-x,n*q);
}
```

We derive a lower bound for $|\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2)|$.

Lemma 3.22. *Let* $x_{\max} \in [2, \infty[$ *and* $x_1, x_2 \in [1, x_{\max}]$ *with* $x_1 \geq x_2 + 1$. *Then*

$$\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2) \leq S_1(x_{\max}) - 2S_2(x_{\max}/2)$$

which means the following lower bound for $|\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2)|$

$$|\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2)| \geq -(S_1(x_{\max}) - 2S_2(x_{\max}/2))$$

Proof. We have

$$\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2) \leq S_1(x_1) - S_2(x_2) - S_2(x_1 - x_2)$$

Differentiation shows that the function $[1, x_1 - 1] \ni x \mapsto S_1(x_1) - S_2(x) - S_2(x_1 - x)$ has a local maximum at $x = x_1/2$ and at most two further points in $[1, x_1 - 1]$ where its derivative is 0, one of these being smaller than $x_1/2$ and the other one being larger than $x_1/2$. Hence

$$\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2) \leq \max\left(S_1(x_1) - 2S_2(x_1/2), S_1(x_1) - S_2(1) - S_2(x_1 - 1)\right)$$

Now differentiation shows that the functions $[2, x_{\max}] \ni x \mapsto S_1(x) - 2S_2(x/2)$ and $[2, x_{\max}] \ni x \mapsto S_1(x) - S_2(1) - S_2(x - 1)$ are increasing. Therefore we get

$$\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2) \leq S_1(x_{\max}) - \min(2S_2(x_{\max}/2), S_2(1) + S_2(x_{\max} - 1))$$

With $f : [2, \infty[\to \mathbb{R}, f(x) := 720 - 2160x - 3240x^2 + 15480x^3 - 17415x^4 + 7965x^5 - 1305x^6$ the inequality $2S_2(x_{\max}/2) \leq S_2(1) + S_2(x_{\max} - 1)$ is equivalent to $f(x_{\max}) \leq 0$. We have $f(x_{\max}) \leq 0$ because of $f(2) = 0, f'(2) = 0, f''(2) < 0, f'''(2) < 0$ and $f''''(x) < 0$ for every $x \in [2, \infty[$. Therefore we get $\min(2S_2(x_{\max}/2), S_2(1) + S_2(x_{\max} - 1)) = 2S_2(x_{\max}/2)$ and with that the proposition. □

The following rather easy inequality could be the foundation of the main theorem of this section.

Lemma 3.23. *Let* $a, b, \tilde{a}, \tilde{b}, \delta_1, \delta_2 \in K$ *with* $ab \leq 0$ *and* $b \in [0, \delta_1], \tilde{b} \in [0, \delta_2]$. *Let* $c \in [0, 1[$ *with* $\mathrm{e}_{\mathrm{rel}}(a, \tilde{a}) \leq c$. *Then*

$$\mathrm{e}_{\mathrm{rel}}(a - b, \tilde{a} - \tilde{b}) \leq c + \max\{\delta_1, \delta_2\}/|a|$$

Proof. We have $|b - \tilde{b}| \leq \max\{\delta_1, \delta_2\}$ and $|a - b - (\tilde{a} - \tilde{b})| \leq |a - \tilde{a}| + |b - \tilde{b}|$. We get $|a - b - (\tilde{a} - \tilde{b})| \leq |a - \tilde{a}| + \max\{\delta_1, \delta_2\}$ and with $|a - b| \geq |a|$ we get the proposition. □

In an example we now want to show how Lemma 3.23 can be used to obtain bounds for the relative error for the value "lc" in Loader's algorithm. In this example we only consider the special case of $\mathrm{e}_{\mathrm{rel}}(x, n \odot \tilde{p}) \leq c$.

Example 3.24. Let $s = 11$ and $t = 52$ so $C_{s,t} = \text{IEEEDouble}$. Let $n_{\max} = 2^{12}$ and $n \in \{2, \ldots, n_{\max}\}, x \in \{1, \ldots, n-1\}, p \in]0,1[, \tilde{p} \in]0,1[\cap F_{s,t}$ with $\mathrm{e}_{\mathrm{rel}}(p, \tilde{p}) \leq 2^{-28} =: c$. We further assume the condition $\mathrm{e}_{\mathrm{rel}}(x, n \odot \tilde{p}) \leq 2^{-20}$ is fulfilled. From Example 3.19 we get that the function `stirlerr` yields $y_1, y_2, y_3 \in C_{s,t}$ with $\mathrm{e}_{\mathrm{rel}}(\mu(n), y_1), \mathrm{e}_{\mathrm{rel}}(\mu(x), y_2), \mathrm{e}_{\mathrm{rel}}(\mu(n-x), y_3)) \leq 2^{-42}$. From this, with Corollary 3.20 and $a := \mu(n) - \mu(x) - \mu(n-x)$ and $\tilde{a} := (y_1 \ominus y_2) \ominus y_3$ we get $\mathrm{e}_{\mathrm{rel}}(a, \tilde{a}) \leq 2 \cdot 10^{-9}$. Lemma 3.22 yields $|a| \geq 6 \cdot 10^{-5}$.

Let $j_{\max} \in \mathbb{N}$ with $\{1, \ldots, 2j_{\max} + 1\} \subseteq C_{s,t}$. Let $y := n \odot \tilde{p}, v := (x \ominus y) \oslash (x \oplus y), s_0 := (x \ominus y) \odot v, e_0 := (2 \odot x) \odot v$ and $e_j := e_{j-1} \odot (v \odot v)$ and $s_j := s_{j-1} \oplus (e_j \oslash (2j+1))$ for $j \in \{1, \ldots, j_{\max}\}$.

Let $\ell \in \{1, \ldots, 20\}$ with $x \in [2^{\ell-1}, 2^{\ell}]$. Because of $\mathrm{e}_{\mathrm{rel}}(x, n \odot \tilde{p}) \leq 2^{-20}$ we have $n \odot \tilde{p} \in [x - 2^{-20+\ell}, x + 2^{-20+\ell}]$.

Let $\alpha := 2^{-40+\ell+1} \oplus 2^{-60+\ell+1}$. From Lemmas 3.10, 3.11 we get

$$0 \leq s_j \leq \alpha$$

for $j \in \{1, \ldots, j_{\max}\}$. From $x \leq 2^{12}$ we get $\ell \leq 12$ and therefore $\alpha \leq 2^{-27} \oplus 2^{-47} \leq 8 \cdot 10^{-9}$. Because of $\mathrm{e}_{\mathrm{rel}}(p, \tilde{p}) \leq c$ and $\mathrm{e}_{\mathrm{rel}}(x, n \odot \tilde{p}) \leq 2^{-20}$ from Lemma 2.29 we get $(n \odot \tilde{p})/x \leq 1 + 2^{-20}$ and therefore

$$\begin{aligned} \mathrm{e}_{\mathrm{rel}}(x, np) &\leq \mathrm{e}_{\mathrm{rel}}(x, n \odot \tilde{p}) + |np - (n \odot \tilde{p})|/x \\ &\leq \mathrm{e}_{\mathrm{rel}}(x, n \odot \tilde{p}) + (n \odot \tilde{p})(1/((1-c)(1-u_t)) - 1)/x \\ &\leq 2^{-20} + (1 + 2^{-20})(1/((1-c)(1-u_t)) - 1) \end{aligned}$$

From Lemma 2.28 we get

$$\mathrm{e}_{\mathrm{rel}}(np, x) \leq \mathrm{e}_{\mathrm{rel}}(x, np)/(1 - \mathrm{e}_{\mathrm{rel}}(x, np))$$

With this bound and $x \leq 2^{12}$ from Lemma 3.12 with $f(x, y) := x \log(x/y) - x + y$ we get

$$0 \leq f(x, y) \leq 2 \cdot 10^{-9}$$

We use Lemma 3.23 and get

$$\mathrm{e}_{\mathrm{rel}}(a - f(k, np), \tilde{a} - s_{j_{\max}}) \leq 2 \cdot 10^{-9} + \frac{8 \cdot 10^{-9}}{6 \cdot 10^{-5}} \leq 2 \cdot 10^{-4}$$

Now we further assume the condition $\mathrm{e}_{\mathrm{rel}}(n - x, n \odot (1 \ominus \tilde{p})) \leq 2^{-20}$ and repeat the above calculations with $n - x$ instead of x and $n \odot (1 \ominus \tilde{p})$ instead of $n \odot \tilde{p}$. Let $y := n \odot (1 \ominus \tilde{p}), w := ((n-x) \ominus y) \oslash ((n-x) \oplus y), \sigma_0 := ((n-x) \ominus y) \odot v, f_0 := (2 \odot (n-x)) \odot v$ and $f_j := f_{j-1} \odot (v \odot v)$ and $\sigma_j := \sigma_{j-1} \oplus (f_j \oslash (2j+1))$ for $j \in \{1, \ldots, j_{\max}\}$. Then again with Lemma 3.23 we get

$$\mathrm{e}_{\mathrm{rel}}(a - f(k, np) - f(n-k, n(1-p)), \tilde{a} - s_{j_{\max}} - \sigma_{j_{\max}}) \leq 2 \cdot 10^{-4} + \frac{8 \cdot 10^{-9}}{6 \cdot 10^{-5}} \leq 4 \cdot 10^{-4}$$

Conclusion

We conclude that we derived intermediate results in the derivation of an accuracy bound for Loader's algorithm for the binomial density. We derived a relative error bound for $\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2)$, a lower bound for $|\mu(x_1) - \mu(x_2) - \mu(x_1 - x_2)|$ and and upper bound for $\mathrm{bd0}(x, np)$ in case of $\mathrm{e_{rel}}(x, np) \leq c$. These results can be combined using the Lemma 3.23.

3.9 Computation of the value "lf" in Loader's algorithm

In this section let $K = \mathbb{R}, v \in [0, 1[$ and $\ell :]0, \infty[\cap C \to C$ with $\ell(x) \in \mathbb{R}$ and $\mathrm{e_{rel}}(\log(x), \ell(x)) \leq v$ for $x \in]0, \infty[\cap C$ with $\log(x) \in R \cup \{0\}$. Futher let $\ell_1 :]-1, 0[\cap C \to C$ with $\ell_1(x) \in \mathbb{R}$ and $\mathrm{e_{rel}}(\log(1 + x), \ell_1(x)) \leq v$ for every $x \in]-1, 0[\cap C$ with $\log(1 + x) \in R$.

We analyse the following function.

```
double lf(double x, double n){
 return M_LN_2PI + log(x) + log1p(- x/n);
}
```

At first we analyse the error propagation of the function $]-1, 0[\ni x \mapsto \log(1 + x)$. We need the following two lemmas.

Lemma 3.25. *Let $x \in]-1, 0[$ and $f :]1 + 1/x, \infty[\to \mathbb{R}, f(\varepsilon) := \log(1+x(1-\varepsilon))/\log(1+x)$. We have $f(0) = 1$. By differentiation we get that f is convex and hence $f(-\varepsilon) - 1 \geq 1 - f(\varepsilon)$ for every $\varepsilon \in [0, -(1 + 1/x)[$. Furthermore, f is decreasing.*

Lemma 3.26. *Let $c \in [0, 1[, f :]-1/(1 + c), 0] \to \mathbb{R}, f(x) := \log(1 + x(1 + c))/\log(1 + x)$. The function f is decreasing.*

Proof. Let $g :]-1/(1 + c), 0] \to \mathbb{R}, g(x) := (1 + c)(1 + x)\log(1 + x) - (1 + x(1 + c))\log(1 + x(1 + c))$. For $x \in]-1/(1 + c), 0]$ we have

$$f'(x) = \frac{(1 + c)\log(1 + x)/(1 + x(1 + c)) - \log(1 + x(1 + c))/(1 + x)}{\log^2(1 + x)}$$

and therefore $f'(x) < 0 \Leftrightarrow g(x) < 0$. We have $g(0) = 0$ and

$$g'(x) = (1 + c)(\log(1 + x) - \log(1 + x(1 + c))) < 0$$

and therefore $g(x) < 0$. □

Lemma 3.27. *Let $\delta \in]0, 1/2]$ and $x \in [-1 + \delta, -\delta], \tilde{x} \in \mathbb{R} \cap C$ and $c \in [0, 1[$ with $\mathrm{e_{rel}}(x, \tilde{x}) \leq c$. Let $(-1 + \delta)(1 + c) > -1$ and $[\log(1 + (-1 + \delta)(1 + c)), \log(1 - \delta(1 - c)] \subseteq R$. Then*

$$\mathrm{e_{rel}}(\log(1 + x), \ell_1(\tilde{x})) \leq \frac{\log(1 + (-1 + \delta)(1 + c))}{\log(\delta)}(1 + v) - 1$$

Proof. Because of $c < 1$ we have $\tilde{x} < -\delta(1-c)$ and $\tilde{x} > (-1+\delta)(1+c)$ and therefore $\log(1+\tilde{x}) \in R$. We have $|\log(1+x) - \ell_1(\tilde{x})| \leq |\log(1+x) - \log(1+\tilde{x})| + v|\log(1+\tilde{x})|$ and with Lemma 3.25 we get

$$\begin{aligned} \mathrm{e}_{\mathrm{rel}}(\log(1+x), \ell_1(\tilde{x})) &\leq \left|1 - \frac{\log(1+\tilde{x})}{\log(1+x)}\right| + v\frac{\log(1+\tilde{x})}{\log(1+x)} \\ &\leq \frac{\log(1+x(1+c))}{\log(1+x)} - 1 + v\frac{\log(1+x(1+c))}{\log(1+x)} \end{aligned}$$

With Lemma 3.26 we get the proposition. □

From the last lemma we get the following corollary.

Corollary 3.28. *Let $s, t \in \mathbb{N}$ and $C = C_{s,t}, R = R_{s,t}$. Let $y_{\max} \in [2, \infty[$ with $y_{\max} \leq 2^{t+1}$ and $y_{\max} \leq 2^{2^{s-1}-2}$. Let $x, y \in [1, y_{\max}] \cap C_{s,t}$ with $x \leq y - 1$. Let*

$$\log(1 + (-1 + 1/y_{\max})(1+u_t)) \geq -2^{2^{s-1}-1} \tag{3.17}$$

$$\log(1 - (1-u_t)/y_{\max}) \leq -2^{2-2^{s-1}} \tag{3.18}$$

Then we get

$$\mathrm{e}_{\mathrm{rel}}(\log(1-x/y), \ell_1(-(x \oslash y)) \leq \frac{\log(y_{\max}/(1-(y_{\max}-1)u_t))}{\log(y_{\max})}(1+v) - 1$$

Proof. Because of $1 \geq x/y \geq 1/y_{\max} \geq 2^{2-2^{s-1}}$ we have $x/y \in R_{s,t}$ and therefore $\mathrm{e}_{\mathrm{rel}}(-x/y, -(x \oslash y)) \leq u_t$. With $\delta := 1/y_{\max} \in]0, 1/2]$ and $c := u_t$ we have $\delta \geq c$ and therefore $(-1+\delta)(1+c) > -1$. We have $-x/y \in [-1+\delta, -\delta]$. Furthermore we have

$$[\log(1 + (-1+\delta)(1+c)), \log(1-\delta(1-c)] \subseteq [-2^{2^{s-1}-1}, -2^{2-2^{s-1}}] \subseteq R_{s,t}$$

Thus we are allowed to apply Lemma 3.27 which because of $1 + (-1+\delta)(1+c) = \delta - c + c\delta = \delta - c(1-\delta) = 1/y_{\max} - u_t(y_{\max}-1)/y_{\max}$ yields the proposed inequality. □

Example 3.29. If $n_{\max} = 2^{40}$ and $u_t = 2^{-53}$ then

$$\frac{\log(1 + (-1+\delta)(1+u_t))}{\log(\delta)}(1+v) - 1 \leq 1.00000441(1+v) - 1$$

Lemma 3.30. *Let $y_{\max} \in [2, \infty[$ and $x, y \in [1, y_{\max}]$ with $x \leq y - 1$. Then*

$$\left|\frac{\log(2\pi x)}{\log(1-x/y)}\right| \geq \frac{\log(2\pi(y_{\max}-1))}{\log(y_{\max})}$$

Proof. Let $f, g : [1, y[\to \mathbb{R}, f(t) := \log(2\pi t)/\log(1 - t/y)$ and $g(t) := t\log(2\pi t) + (y - t)\log(1 - t/y)$. Differentiation of f yields that $f' > 0 \Leftrightarrow g > 0$. We have $g'(t) = \log(2\pi t) - \log(1 - t/y) > 0$ for $t \in [1, y[$ and $g(1) = \log(2\pi) + (y - 1)\log(1 - 1/y) \geq \log(2\pi) + (y - 1)\frac{-1/y}{1-1/y} = \log(2\pi) - 1 > 0$. Therefore $g > 0$. Hence f increases and $f(x) \leq f(y - 1) = -\log(2\pi(y - 1))/\log(y)$. Let $h, k : [2, y_{\max}] \to \mathbb{R}, h(t) := -\log(2\pi(t - 1))/\log(t)$ and $k(t) := (t - 1)\log(2\pi(t - 1)) - t\log(t)$. Then $h' > 0 \Leftrightarrow k > 0$. We have $k'(t) = \log(2\pi(t - 1)) - \log(t) > 0$ for $t \in [2, y_{\max}]$ and $k(2) = \log(2\pi) - \log(4) > 0$. Therefore h is increasing and we get $f(x) \leq f(y - 1) = h(y) \leq h(y_{\max}) = -\frac{\log(2\pi(y_{\max}-1))}{\log(y_{\max})} \leq 0$. From that we get the proposition. □

Corollary 3.31. *Let $s, t \in \mathbb{N}, C = C_{s,t}, R = R_{s,t}$. Let $y_{\max} \in [2, \infty[$ with $y_{\max} \leq 2^{t+1}$ and $y_{\max} \leq 2^{2^{s-1}-2}$ and $x, y \in [1, y_{\max}] \cap C_{s,t}$ with $x \leq y - 1$. Let $a \in \mathbb{R} \cap C_{s,t}$ with $\mathrm{e}_{\mathrm{rel}}(\log(2\pi), a) \leq u_t$ and $b := a \oplus \ell(x)$. We assume* (3.17)*,* (3.18) *and*

$$(1 + u_t)\log(2\pi) + (1 + v)\log(y_{\max} - 1) \leq 2^{2^{s-1}-1}$$

Let

$$c_1 := (1 + u_t)(1 + \max(v, u_t)) - 1$$

$$c_2 := \frac{\log(y_{\max}/(1 - (y_{\max} - 1)u_t))}{\log(y_{\max})}(1 + v) - 1$$

$$c_3 := \log(2\pi(y_{\max} - 1))/\log(y_{\max})$$

Let $c_2 < 1$ and

$$(1 - c_1)\log(2\pi(y_{\max} - 1)) + (1 + c_2)\log(1/y_{\max}) \geq 2^{2-2^{s-1}}$$

Then we get

$$\mathrm{e}_{\mathrm{rel}}(\log(2\pi x(1 - x/y), b \oplus \ell_1(-(x \oslash y))) \leq (1 + u_t)\left(1 + \frac{c_1}{1 - c_3^{-1}} + \frac{c_2}{c_3 - 1}\right) - 1$$

Proof. Because of $\log(x) \leq x \in C_{s,t}$ we have $\log(x) \in R \cup \{0\}$ and therefore $\mathrm{e}_{\mathrm{rel}}(\log(x), \ell(x)) \leq v$. Because of $a + \ell(x) \leq (1 + u_t)\log(2\pi) + (1 + v)\log(y_{\max} - 1) \leq 2^{2^{s-1}-1}$ we have $a + \ell(x) \in R_{s,t}$ and with Lemma 2.36 we get $\mathrm{e}_{\mathrm{rel}}(\log(2\pi x), b) \leq c_1$. From Corollary 3.28 we get $\mathrm{e}_{\mathrm{rel}}(\log(1 - x/y), \ell_1(-(x \oslash y)) \leq c_2$. Therefore we have $b + \ell_1(-(x \oslash y)) \geq (1 - c_1)\log(2\pi y) + (1 + c_2)\log(1 - x/y)$. Now differentiation yields $b + \ell_1(-(x \oslash y)) \geq (1 - c_1)\log(2\pi(y_{\max} - 1)) + (1 + c_2)\log(1/y_{\max}) \geq 2^{2-2^{s-1}}$. Because of $c_2 < 1$ and $\log(1 - x/y) < 0$ we also have $\ell_1(-(x \oslash y)) < 0$ and therefore $b + \ell_1(-(x \oslash y)) < b \in \mathbb{R} \cap C_{s,t}$. Hence $b + \ell_1(-(x \oslash y)) \in R_{s,t}$. From Lemma 3.30 we get $|\log(2\pi x)|/|\log(1 - x/y)| \geq c_3$. We use Lemma 2.38 and get the proposition. □

Example 3.32. For $s = 11, t = 52, v = 2^{-20}, n_{\max} = 2^{40}$ and $n \in \{2, \ldots, n_{\max}\}, x \in \{1, \ldots, n - 1\}$ and $a \in \mathbb{R} \cap C_{s,t}$ with $\mathrm{e}_{\mathrm{rel}}(\log(2\pi), a) \leq u_t$ we get

$$\mathrm{e}_{\mathrm{rel}}(\log(2\pi x(1 - x/n), (a \oplus \ell(x)) \oplus \ell_1(-(x \oslash n))) \leq 0.0001006$$

3.10 Error propagation in exponentiation

In this section let $K = \mathbb{R}$.

Lemma 3.33. *Let $c \in [0, \infty[$ and $x \in [-c, c]$. Then we have*

$$|1 - \exp(x)| \leq \exp(c) - 1$$

Proof. Because $\exp$ has positive derivative we get $\exp(x) \in [\exp(-c), \exp(c)]$ and therefore

$$|1 - \exp(x)| \leq \max\{\exp(c) - 1, 1 - \exp(-c)\}$$

We now show $\exp(c) - 1 \geq 1 - \exp(-c)$. Let $f : [0, \infty[\to \mathbb{R}$ defined by $f(y) := \exp(y) + \exp(-y)$. Then $f'(y) = \exp(y) - \exp(-y)$ and we get $f'(y) = 0 \Leftrightarrow y = 0$. Because of $f''(y) = \exp(y) + \exp(-y) > 0$ we get that $y = 0$ is minimum of f and therefore $\exp(c) + \exp(-c) = f(c) \geq f(0) = 2$. From this we get $\exp(c) - 1 \geq 1 - \exp(-c)$ and therefore

$$|1 - \exp(x)| \leq \max\{\exp(c) - 1, 1 - \exp(-c)\} = \exp(c) - 1$$

□

Lemma 3.34. *Let $f :]-\infty, 0[\cap C \to C$ and $v \in [0, 1[$ with $f(x) \in \mathbb{R}$ and $\mathrm{e}_{\mathrm{rel}}(\exp(x), f(x)) \leq v$ for every $x \in]-\infty, 0[\cap C$ with $\exp(x) \in R$. Let $x \in]-\infty, 0[, \tilde{x} \in]-\infty, 0[\cap C, c \in [0, 1[$ with $\exp(\tilde{x}) \in R$ and $\mathrm{e}_{\mathrm{rel}}(x, \tilde{x}) \leq c$ Then*

$$\mathrm{e}_{\mathrm{rel}}(\exp(x), f(\tilde{x})) \leq \exp(c|x|)(1 + v) - 1$$

Proof. We have

$$\begin{aligned} \frac{|\exp(x) - f(\tilde{x})|}{\exp(x)} &\leq \frac{|\exp(x) - \exp(\tilde{x})|}{\exp(x)} + v \exp(\tilde{x}) / \exp(x) \\ &= |1 - \exp(\tilde{x} - x)| + v \exp(\tilde{x} - x) \\ &\leq \exp(c|x|)(1 + v) - 1 \end{aligned}$$

In the last step $|x - \tilde{x}| \leq c|x|$ and Lemma 3.33 was used. □

Example: If $s, t \in \mathbb{N}, C = C_{s,t}$ and $x \geq \log(2^{2-2^{s-1}})/(1 + c)$ then $\exp(\tilde{x}) \geq 2^{2-2^{s-1}}$ and $|x| \leq (2^{s-1} - 2)\log(2)$ and therefore

$$\exp(c|x|)(1 + v) - 1 \leq \exp(c(2^{s-1} - 2)\log(2))(1 + v) - 1 = 2^{c(2^{s-1}-2)}(1 + v) - 1$$

Example: If $s = 11, c = 2^{-4}, v = 2^{-20}$ then we get $\exp(c(2^{s-1} - 2)\log(2))(1 + v) - 1 \leq 0.074$

Chapter 4

Computations of rigorous bounds for binomial, multinomial and multivariate hypergeometrical probabilities

We still consider the computation of Scan Probabilities for Markov increments. Instead of using the "rounding to nearest mode" to get approximate values and then use the error bounds derived in the last chapter, we are able to compute rigorous bounds using the functions $\overline{\oplus}, \underline{\oplus}, \overline{\ominus}, \ldots, \underline{\oslash}$ which we defined in chapter 2. A case study can be found in [6].

4.1 Displaying double precision floating point numbers in hexadecimal format and as rational expression

The following code is suitable for displaying a double precision floating point number `x` in C programs in hexadecimal format.

```
printf("%p\n",x);
```

This prints a sequence of 8 hexadecimal characters which represents the floating point number. The first 3 characters of that sequence represent the sign s and the exponent e in the representation

$$x = (-1)^s(1 + d \cdot 2^{-52})2^{e-1023}$$

while the last 5 characters represent the mantissa d. The hexadecimal sequence can be converted into a rational representation with Mathematica, which is described below. In Mathematica the command `16^^` can be used to convert a hexadecimal number into a decimal number. For example the hexadecimal number `AA` with Mathematica can be converted using the command `16^^AA` and Mathematica returns the value 170 as result. With the help of this command, a hexadecimal

represenation of a double precision floating point number which is displayed by the C instruction `printf` can be converted with Mathematica into a rational representation. For example the C instruction `printf("%p\n", 0.1);` displays the hexadecimal number `3FB999999999999A`.

This can be converted into a rational expression with Mathematica with the following command:

```
x= (1 + 2^-52*16^^999999999999A)*2^(16^^3FB - 1023)
```

As result Mathematica provides the rational representation $\frac{3602879701896397}{36028797018963968}$ of the floating point number `0.1` that we entered to the function `printf`.

Negative double precision floating point numbers are characterised by a hexadecimal representation in which the first character is larger than or equal to 8. In negative numbers, the hexadecimal representation of the absolute value can be obtained by subtracting the hexadecimal character 8 from the first character of the hexadecimal sequence which represents the floating point number. For example `printf("%p\n", -0.1);` displays the hexadecimal number `BFB999999999999A`. Here the difference of the hexadecimal value `B` and `8` is `3`, so the hexadecimal representation of the absolute value of `-0.1` is `3FB999999999999A`.

4.2 Changing the rounding mode in C programs

The rounding mode in C according to the C 99 standard can be changed using the

```
#include<fenv.h>
```

header file and the commands

```
fesetround(FE_DOWNWARD);
```

for rounding downwards, and

```
fesetround(FE_UPWARD);
```

for rounding upwards.

The following programs measure the time needed to change the rounding mode.

```
void measureTimeAddition(void){
double z=1.0;
double eps=pow(2,-52);
int i;
for(i=0;i<2147483647;i++){
z=z+eps;
}
}
```

The execution of the program `measureTimeAddition` took 5.3 seconds.

```
void measureTimeChangeRounding(void){
double z=1.0;
double eps=pow(2,-52);
int i;
for(i=0;i<2147483647;i++){
fesetround(FE_UPWARD);
fesetround(FE_DOWNWARD);
}
}
```

The execution of the program `measureTimeChangeRounding` took 269.8 seconds.

```
void measureTimeNoOperation(void){
double z=1.0;
double eps=pow(2,-52);
int i;
for(i=0;i<2147483647;i++){
}
}
```

The execution of the program `measureTimeNoOperation` took 3.2 seconds.

In this experiment changing the rounding mode took about 63 times the computation time for an addition.

4.2.1 Example: Computation of rigorous bounds for binomial probabilities

The following function can be used to compute lower and upper bounds for the binomial density $\mathrm{b}_{n,p}(k)$. Input variables are n, k and a lower bound `lowerp` and an upper bound `upperp` for the value p. If the input variable `rounding` is 0, then the function `bin` returns an upper bound for $\mathrm{b}_{n,p}(k)$, otherwise the function returns a lower bound for $\mathrm{b}_{n,p}(k)$.

```
#include<stdio.h>
#include<fenv.h>

double bin(int n, int k, double lowerp, double upperp, int rounding){
int d=n-k;
double e;
e=1.0;
```

```
int i;

if(rounding==0){
fesetround(FE_UPWARD);
for(i=1;i<=k;i++){
e=e*(double)(d+i)/(double)i;
e=e*upperp;
}
for(i=1;i<=d;i++){
e=e*(1-lowerp);
}
}
else{
fesetround(FE_DOWNWARD);
for(i=1;i<=k;i++){
e=e*(double)(d+i)/(double)i;
e=e*lowerp;
}
for(i=1;i<=d;i++){
e=e*(1-upperp);
}
}
return e;
}
```

Example: The following program computes lower and upper bounds for the binomial probability $\mathrm{b}_{30,2/3}(20)$

```
int main (void){
double b=2.0;
double c=3.0;
fesetround(FE_UPWARD);
double upperp=b/c;
fesetround(FE_DOWNWARD);
double lowerp=b/c;
printf("%p\n",  bin(30,20,lowerp,upperp,0));
printf("%p\n",  bin(30,20,lowerp,upperp,1));
return 0;
}
```

The program returns the lower bound `3FC39600E4A68EB8` and the upper bound `3FC39600E4A68F0C` for the binomial probability $\mathrm{b}_{30,2/3}(20)$. With Mathematica we are able to compute the rational forms of these hexadecimal representation:

Rational form of the lower bound 3FC39600E4A68EB8:

```
(1 + 2^-52*16^^39600E4A68EB8)*2^(16^^3FC - 1023)
```

This gives us the exact rational expression $\frac{689119392223703}{4503599627370496}$ as lower bound for $\mathrm{b}_{30,2/3}(20)$.
Rational form of the upper bound 3FC39600E4A68F0C:

```
(1 + 2^-52*16^^39600E4A68F0C)*2^(16^^3FC - 1023)
```

This gives us the exact rational expression $\frac{1378238784447427}{9007199254740992}$ as upper bound for $\mathrm{b}_{30,2/3}(20)$.

4.3 Computation of rigorous bounds for rectangle probabilities for a multinomially distributed random variable

In the last section we stated functions which are suitable for computing rigorous bounds for the binomial density. These functions can be used to further compute rigorous bounds for rectangle probabilities for a multinomially distributed random variable. The following algorithm is an efficient C implementation of the algorithm which was stated in the Appendix A of [5]. Here the so called "multiplication method" which was stated in Appendix B of Loader [16] was used.

In this example, the algorithm computes a rigorous upper bound for the probability $\mathbb{P}(N_1 \in \{j_1, \ldots, k_1\}, \ldots, N_d \in \{j_d, \ldots, k_d\})$ with $(N_1, \ldots, N_d) \sim \mathrm{M}_{n,(1/d,\ldots,1/d)}$ and $n = 10, d = 6, j_1 = \ldots = j_d = 0, k_1 = \ldots = k_d = 4$.

```
#include <stdio.h>
#include <fenv.h>
#include <stdlib.h>

#define max( a, b ) ( ((a) > (b)) ? (a) : (b) )
#define min( a, b ) ( ((a) < (b)) ? (a) : (b) )

void sum(int n, double* startadress, double* sum){
int i;
*sum=0;
for (i=0; i<n; i++){ *sum=*sum + *(startadress+i);}
}


void isum(int n, int* startadress, int* sum){
int i;
*sum=0;
```

```
for (i=0; i<n; i++){ *sum=*sum + *(startadress+i);}
}

double upperbnp(int k,int n, double pu, double po){
fesetround(FE_UPWARD);
if (2*k>n){
double ponew,punew;
ponew=1-pu;
fesetround(FE_DOWNWARD);
punew=1-po;
return(upperbnp(n-k,n,punew,ponew));
}
double f=1.0;
int j0=0,j1=0,j2=0;
while ((j0<k)| (j1<k)|(j2<n-k))
{ if((j0<k)&& (f<1))
{j0++;
f*= (double)(n-k+j0)/(double)j0;
}
else
{if (j1<k){j1++;f*= po;}
else { j2++; f*= 1-pu;}
}
}
return(f);
}

double upperMarkovtransition (int k, int i, int j, double* pu,
 double* po, int d, int n){
double so,su;
fesetround(FE_UPWARD);
sum(d-k+1,&po[k],&so);
fesetround(FE_DOWNWARD);
sum(d-k+1,&pu[k],&su);
double psu=pu[k]/so;
fesetround(FE_UPWARD);
double pso=po[k]/su;
double prob=upperbnp(j-i,n-i,psu,pso);
return prob;
}

double upperStartProb(int i,int n, double* pu, double* po){
return upperbnp(i,n,pu[0],po[0]);
```

```
}

void upperRectangleProb(void){
int i,k;
int n = 10;
int d = 6;
double zahler=1.0;
double nenner=(double) d;
double* pu =  (double*) malloc (d * sizeof(double));
double* po =  (double*) malloc (d * sizeof(double));
fesetround(FE_DOWNWARD);
for (i=0;i<d;i++){ pu[i]=zahler/nenner; }
fesetround(FE_UPWARD);
for (i=0;i<d;i++){ po[i]=zahler/nenner; }
int* b =  (int*) malloc (d * sizeof(int));
for (i=0;i<d;i++){ b[i]=0; }
int* c =  (int*) malloc (d * sizeof(int));
for (i=0;i<d;i++){ c[i]=4; }

int* alpha =  (int*) malloc ( (d-1) * sizeof(int));
int* beta =  (int*) malloc ( (d-1) * sizeof(int));
int z1,z2;
for (k =1;k<d;k++){
 isum(d-k,&c[k],&z1);
 isum(k,&b[0],&z2);
 alpha[k-1]= max(n-z1,z2);
 }
for (k =1;k<d;k++){
 isum(d-k,&b[k],&z1);
 isum(k,&c[0],&z2);
 beta[k-1]= min(n-z1,z2);
 }


double* temp =  (double*) malloc ( (n+1) * sizeof(double));
double* P =  (double*) malloc ( (n+1) * sizeof(double));
double* R =  (double*) malloc ( (n+1) * sizeof(double));
double* Q;
int j;

for (j=0;j<=n;j++){ R[j]=0; }
for (j=0;j<=n;j++){ P[j]=0; }
```

```
for ( j=alpha[0]; j<= beta[0]; j++) { P[j]= upperStartProb(j,n,pu,po); }



int x,su,so;
for (k = 2; k<d; k++) {

Q=R;

    for (x = alpha[k-1]; x <= beta[k-1]; x ++) {

    su=max(x-c[k-1],alpha[k-2]);
    so=min(x-b[k-1],beta[k-2]);

if (su<= so){
for (j=su;j<= so; j++) {
temp[j] = upperMarkovtransition(k-1,j,x,pu,po,d,n)*P[j];};
sum(so-su+1,&temp[su],&Q[x]);
}


}

for (j=0;j<=n;j++){ P[j]=0;}
R=P;
P=Q;
}

double result;
sum(n+1,&P[0],&result);
printf("%.20f",result);
}



int main (void){
upperRectangleProb();
return 0;
}
```

4.4 Comparison of the multiplication method and Loader's algorithm for the binomial density

While the multiplication method for the binomial density allows the computation of rigorous bounds when the rounding modes of the computer are changed, it is much slower in comparison with Loader's algorithm for the binomial density. For example, we consider the computation of the values $\mathbb{P}(N_1, \ldots, N_d \leq k)$ of the cumulative distribution function for a multinomially distribed random variable $(N_1, \ldots, N_d) \sim \mathrm{M}_{n,p}$ with $n, d \in \mathbb{N}, d \geq 2$ and $p = (1/d, \ldots, 1/d)$. Table 4.1 lists the times needed to compute these values with the function stated in Appendix A, when the binomial transition probabilities were computed with one of the two different methods. All computations were done on a 3.7 GHz CPU with 4.0 GB Ram.

Table 4.1: Time needed to compute $\mathbb{P}(N_1, \ldots, N_d \leq k)$ for $(N_1, \ldots, N_d) \sim \mathrm{M}_{n,p}$

n	d	k	$\mathbb{P}(N_1, \ldots, N_d \leq k)$	Time (multiplication method)	Time (Loader's algorithm)
100	100	4	0.7016461	1.0 s	0.25 s
100	100	5	0.9475989	1.3 s	0.33 s
100	100	6	0.9929082	1.6 s	0.36 s
300	250	4	0.1332788	17.5 s	1.7 s
300	250	5	0.6913766	22.6 s	2.1 s
300	250	6	0.9417305	29.2 s	2.5 s
500	250	5	0.0111244	47.3 s	2.8 s
500	250	6	0.3171264	61.1 s	3.5 s
500	250	7	0.7644753	75.6 s	4.1 s

4.5 Computation of rigorous bounds for rectangle scan probabilities for a multinomially distributed random variable

Let $(N_1, \ldots, N_d) \sim \mathrm{M}_{n,p}$ with $n = 500, d = 365$ and $p = (1/d, \ldots, 1/d)$. Let

$$S := \max_{k=1}^{d-2}(N_k + N_{k+1} + N_{k+2})$$

In Appendix B we listed an implementation of the Algorithm A from chapter 1 which computes lower bounds and upper bounds for the values $\mathbb{P}(S \leq k)$ of the cumulative distribution function of S for $k \in \{8, \ldots, 15\}$. These values are listed in Table 4.2.

Table 4.2: Bounds for the cumulative distribution function of the multinomial scan S

k	$\mathbb{P}(S \leq k)$	k	$\mathbb{P}(S \leq k)$	k	$\mathbb{P}(S \leq k)$	k	$\mathbb{P}(S \leq k)$
8	0.0007795	10	0.3773	12	0.9030	14	0.9920
	0.0007796		0.3774		0.9031		0.9921
9	0.0661	11	0.7210	13	0.9708	15	0.9979
	0.0662		0.7211		0.9709		0.9980

4.5.1 Comparison of the accuracy of bounds in double precision and in single precision

Following [6], we do a following case study which compares the accuracy of rigorous bounds for multinomial scan probabilities when either the double precision number system $\mathrm{IEEEDouble}$ or the single precision number system $\mathrm{IEEESingle}$ are used.

For a quantitative analysis of the accuracy of computed probabilities we need to consider absolute and relative errors. For $p, \tilde{p} \in [0,1]$ we define the **absolute error**

$$e_{\mathrm{abs}}(p, \tilde{p}) := |p - \tilde{p}|$$

and the **relative error**

$$e_{\mathrm{rel}}(p, \tilde{p}) := \max\left\{\frac{e_{\mathrm{abs}}(p, \tilde{p})}{p}, \frac{e_{\mathrm{abs}}(1-p, 1-\tilde{p})}{1-p}\right\} = \frac{|p - \tilde{p}|}{\min(p, 1-p)}$$

in the approximation of p by $\tilde{p}$, with $\frac{0}{0} := 0$ and $\frac{x}{0} := \infty$ for $x > 0$. For $a, b \in [0,1]$ with $a \leq b$ and $\tilde{p} \in [a, b]$ we further define the **absolute error**

$$e_{\mathrm{abs}}([a,b], \tilde{p}) := \max_{p \in [a,b]} e_{\mathrm{abs}}(p, \tilde{p}) = \max\{b - \tilde{p}, \tilde{p} - a\}$$

and the **relative error**

$$e_{\mathrm{rel}}([a,b], \tilde{p}) := \max_{p \in [a,b]} e_{\mathrm{rel}}(p, \tilde{p})$$

in the approximation of a probability which is known to lie in $[a, b]$ by $\tilde{p}$. We get simple formulas for $e_{\mathrm{rel}}([a,b], \tilde{p})$ in the following two cases. If $a, b \in [0, 1/2]$ or $a, b \in [1/2, 1]$ we have

$$e_{\mathrm{rel}}([a,b], \tilde{p}) = \max\{e_{\mathrm{rel}}(a, \tilde{p}), e_{\mathrm{rel}}(b, \tilde{p})\}$$

Hence, if $a, b \in]0, 1/2]$ we have

$$e_{\mathrm{rel}}([a,b], \tilde{p}) = \max\{\frac{\tilde{p} - a}{a}, \frac{b - \tilde{p}}{b}\}$$

and if $a, b \in [1/2, 1[$ we have

$$e_{\mathrm{rel}}([a,b], \tilde{p}) = \max\{\frac{\tilde{p} - a}{1-a}, \frac{b - \tilde{p}}{1-b}\}$$

For accuracy measurements in interval calculations we use the following mini-max errors.

Definition 4.1. For $a, b \in [0,1]$ with $a \leq b$ we define the **absolute error**

$$e_{\mathrm{abs}}([a,b]) := \min_{\tilde{p} \in [a,b]} e_{\mathrm{abs}}([a,b], \tilde{p}) = e_{\mathrm{abs}}([a,b], \frac{a+b}{2}) = \frac{b-a}{2}$$

and the **relative error**

$$e_{\mathrm{rel}}([a,b]) := \min_{\tilde{p} \in [a,b]} e_{\mathrm{rel}}([a,b], \tilde{p})$$

in the approximation of a probability by the interval $[a, b]$.

Easy calculations yield the following formulas:

Theorem 4.2. *If* $a, b \in [0, 1/2]$ *we have*

$$\forall \tilde{p} \in [a, b] : e_{\mathrm{rel}}([a, b], \tilde{p}) \leq e_{\mathrm{rel}}([a, b], \frac{2ab}{a + b}) = \frac{b - a}{b + a}$$

Hence

$$e_{\mathrm{rel}}([a, b]) = \frac{b - a}{b + a}$$

If $a, b \in [1/2, 1]$ *we have*

$$\forall \tilde{p} \in [a, b] : e_{\mathrm{rel}}([a, b], \tilde{p}) \leq e_{\mathrm{rel}}([a, b], \frac{a + b - 2ab}{2 - a - b}) = \frac{b - a}{2 - a - b}$$

Hence

$$e_{\mathrm{rel}}([a, b]) = \frac{b - a}{2 - a - b}$$

Note that the absolute error $e_{\mathrm{abs}}([a, b])$ and the relative error $e_{\mathrm{rel}}([a, b])$ need not be reached simultaneously by one of the approximators. It need not be reached at all, as the following example illustrates.

Example 4.3. In Table 4.3 we listed the errors $e_{\mathrm{abs}}([a, b], \tilde{p})$ and $e_{\mathrm{rel}}([a, b], \tilde{p})$ for $[a, b] = [0.02, 0.03]$ and different approximators $\tilde{p}$. We see that $e_{\mathrm{abs}}([a, b]) = 0.005$ and $e_{\mathrm{rel}}([a, b]) = 1/5$. If we take the upper bound $\tilde{p} = b$ as approximator for the unknown probability p , neither $e_{\mathrm{abs}}([a, b], \tilde{p}) = e_{\mathrm{abs}}([a, b])$ is reached, nor $e_{\mathrm{rel}}([a, b], \tilde{p}) = e_{\mathrm{rel}}([a, b])$. If, for example, the unknown probability is

Table 4.3: Errors for the interval $[a, b] = [0.02, 0.03]$

$\tilde{p}$	$e_{\mathrm{abs}}([a, b], \tilde{p})$	$e_{\mathrm{rel}}([a, b], \tilde{p})$
$2ab/(a + b) = 0.024$	0.006	$\mathbf{1/5}$
$(a + b)/2 = 0.025$	$\mathbf{0.005}$	$1/4$
a	0.01	$1/3$
b	0.01	$1/2$

$p = (3/10)^3 = 0.027$, then the errors are as listed in Table 4.4.

Table 4.4: Errors for the probability $p = 0.027$

$\tilde{p}$	$e_{\mathrm{abs}}(p,\tilde{p})$	$e_{\mathrm{rel}}(p,\tilde{p})$
0.024	0.003	$3/27$
0.025	0.002	$2/27$
a	0.007	$7/27$
b	0.003	$3/27$

Examples

For $N \sim \mathrm{M}_{n,p}$ with $n = 500$, $d = 365$, $p = (1/d, \ldots, 1/d)$ and $k \in \{4, \ldots, 32\}$ we computed an upper bound $\overline{p}$ and a lower bound $\underline{p}$ for the probability $\mathbb{P}(\max_{i=1}^{d-2}(N_i + N_{i+1} + N_{i+2}) \leq k)$ with the Algorithm from Appendix B . In Table 4.5 we tabulate the computed bounds $\underline{p}, \overline{p}$ and analyze their accuracy in double precision, in Table 4.6 we list the results if all computations are done in single precision. Numbers written in typewriter font are hexadecimal. The coloumn titled "approx" gives the known decimal digits of a value of the "probability representation number system" T, that lies nearest to the exact value. The probability representation number system T consists of all numbers with 7 decimal digits without leading zeros or nines. We use the notation $.0^x$ as an abbreviation for a decimal point followed by x zeros, analogously $.9^x$. The symbol ? appearing in a number means that the following digits are not exactly known.

The value e_{abs} resp. e_{rel} is the minimal upper bound for $e_{\mathrm{abs}}([\underline{p}, \overline{p}])$ resp. $e_{\mathrm{rel}}([\underline{p}, \overline{p}])$ which has the form $c \cdot 10^k$ where c has 3 significant digits and $k \in \mathbb{Z}$.

Thus, in Table 4.5 the line with $k = 15$ means that the probability $\mathbb{P}(\max_{i=1}^{d-2}(N_i + N_{i+1} + N_{i+2}) \leq 15)$ lies in the interval $[\underline{p}, \overline{p}]$ with

$$\begin{aligned}
\overline{p} &= \texttt{1.fef956911fe58} \cdot 2^{-1} \\
&= (1 + 15 \cdot 16^{-1} + 14 \cdot 16^{-2} + \ldots + 8 \cdot 16^{-13}) \cdot 2^{-1} \\
&= 0.997996049132733098474545840872451663017272949218 75 \\
\underline{p} &= \texttt{1.fef95690c7eda} \cdot 2^{-1} \\
&= (1 + 15 \cdot 16^{-1} + \ldots + 10 \cdot 16^{-13}) \cdot 2^{-1} \\
&= 0.997996049092729764495857125439215451478958129882 8125
\end{aligned}$$

with all equalities exact. The minimal upper bound for $e_{\mathrm{abs}}([\underline{p}, \overline{p}])$ which has the form $c \cdot 10^k$ where c has 3 significant digits and $k \in \mathbb{Z}$ is $2.01 \cdot 10^{-11}$ and the minimal upper bound for $e_{\mathrm{abs}}([\underline{p}, \overline{p}])$ which has this form is $9.99 \cdot 10^{-9}$. A value of the number system T which is nearest to the exact probability is 0.9979961. As the numbers of the system T in the interval $[0.001, 0.9989999]$ differ by 10^{-7}, just knowing the approximate value we can infer that the absolute error in this approximation is less than 10^{-7}.

The computed probabilities can be used as p-values for tests that check data on clusters. For example: Let $n = 500$ patients arrive at a clinic in $d = 365$ days. We compute the probability that there exist three successive days in which together more than 15 patients arrive. From the

Table 4.5: Upper and lower bounds $\overline{p}, \underline{p}$ for $\mathbb{P}(\max_{i=1}^{d-2} N_i + N_{i+1} + N_{i+2} \leq k)$ with $N \sim \mathrm{M}_{n,p}$, $n = 500$, $d = 365$, $p = (1/d, \ldots, 1/d)$ and $k \in \{4, \ldots, 32\}$.

k	$\underline{p}, \overline{p}$	e_{abs}	e_{rel}	approx
4	0	0	0	**0**
5	`1.1c5df1e1a1f83` $\cdot 2^{-178}$ `1.1c5df1e171043` $\cdot 2^{-178}$	$5.82 \cdot 10^{-65}$	$2.01 \cdot 10^{-11}$	$\mathbf{.0^{53}28993}$
6	`1.b826f22f10057` $\cdot 2^{-67}$ `1.b826f22ec43c3` $\cdot 2^{-67}$	$2.34 \cdot 10^{-31}$	$2.01 \cdot 10^{-11}$	$\mathbf{.0^{19}11651}$
7	`1.b71c492587c97` $\cdot 2^{-27}$ `1.b71c49253c2df` $\cdot 2^{-27}$	$2.57 \cdot 10^{-19}$	$2.01 \cdot 10^{-11}$	$\mathbf{.0^{7}12780}$
8	`1.98b8351d76fbd` $\cdot 2^{-11}$ `1.98b8351d309cf` $\cdot 2^{-11}$	$1.57 \cdot 10^{-14}$	$2.01 \cdot 10^{-11}$	$\mathbf{.0^{3}77957}$
9	`1.0f0230ce6f8a1` $\cdot 2^{-4}$ `1.0f0230ce40e15` $\cdot 2^{-4}$	$1.33 \cdot 10^{-12}$	$2.01 \cdot 10^{-11}$	**.0661642**
10	`1.826e2adb7befd` $\cdot 2^{-2}$ `1.826e2adb39686` $\cdot 2^{-2}$	$7.57 \cdot 10^{-12}$	$2.01 \cdot 10^{-11}$	**.3773734**
11	`1.7131cf887a229` $\cdot 2^{-1}$ `1.7131cf883a935` $\cdot 2^{-1}$	$1.45 \cdot 10^{-11}$	$5.19 \cdot 10^{-11}$	**.7210832**
12	`1.ce576094ddb84` $\cdot 2^{-1}$ `1.ce5760948e1f6` $\cdot 2^{-1}$	$1.81 \cdot 10^{-11}$	$1.87 \cdot 10^{-10}$	**.9030104**
13	`1.f1162301d80ec` $\cdot 2^{-1}$ `1.f1162301827ae` $\cdot 2^{-1}$	$1.95 \cdot 10^{-11}$	$6.69 \cdot 10^{-10}$	**.9708720**
14	`1.fbef9498b0df9` $\cdot 2^{-1}$ `1.fbef9498596d7` $\cdot 2^{-1}$	$1.99 \cdot 10^{-11}$	$2.51 \cdot 10^{-9}$	**.9920622**
15	`1.fef956911fe58` $\cdot 2^{-1}$ `1.fef95690c7eda` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$9.99 \cdot 10^{-9}$	**.9979961**
16	`1.ffc1fbbfd6e58` $\cdot 2^{-1}$ `1.ffc1fbbf7ecb1` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$4.24 \cdot 10^{-8}$	$\mathbf{.9^{3}52685}$
17	`1.fff23b0d23a3c` $\cdot 2^{-1}$ `1.fff23b0ccb810` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$1.91 \cdot 10^{-7}$	$\mathbf{.9^{3}89495}$
18	`1.fffd1d22cb527` $\cdot 2^{-1}$ `1.fffd1d22732da` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$9.11 \cdot 10^{-7}$	$\mathbf{.9^{4}77980}$
19	`1.ffff6d5024936` $\cdot 2^{-1}$ `1.ffff6d4fcc6e4` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$4.59 \cdot 10^{-6}$	$\mathbf{.9^{5}56284}$
20	`1.ffffe4570f39a` $\cdot 2^{-1}$ `1.ffffe456b7146` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$2.44 \cdot 10^{-5}$	$\mathbf{.9^{6}17567}$
21	`1.fffffb08bd13c` $\cdot 2^{-1}$ `1.fffffb0864ee9` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$1.36 \cdot 10^{-4}$	$\mathbf{.9^{6}8520}$?
22	`1.ffffff264f47d` $\cdot 2^{-1}$ `1.ffffff25f7228` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$7.91 \cdot 10^{-4}$	$\mathbf{.9^{7}74}$?
23	`1.fffffdc79315` $\cdot 2^{-1}$ `1.fffffdc210c0` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$4.83 \cdot 10^{-3}$	$\mathbf{.9^{8}6}$?
24	`1.fffffffa913ba` $\cdot 2^{-1}$ `1.fffffffa39167` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$3.08 \cdot 10^{-2}$	$\mathbf{.9^{9}}$?
25	`1.ffffffff53a50` $\cdot 2^{-1}$ `1.ffffffefb7fe` $\cdot 2^{-1}$	$2.01 \cdot 10^{-11}$	$2.04 \cdot 10^{-1}$	$\mathbf{.9^{9}}$?
26	1 `1.fffffffb44b7` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$\mathbf{.9^{10}}$?
27	1 `1.fffffffcf373` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$\mathbf{.9^{10}}$?
28	1 `1.fffffffd2fd3` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$\mathbf{.9^{10}}$?
29	1 `1.fffffffd37fa` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$\mathbf{.9^{10}}$?
30	1 `1.fffffffd3908` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$\mathbf{.9^{10}}$?
31	1 `1.fffffffd392a` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$\mathbf{.9^{10}}$?
32	1 `1.fffffffd392a` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$\mathbf{.9^{10}}$?

Table 4.6: Upper and lower bounds $\overline{p}, \underline{p}$ for $\mathbb{P}(\max_{i=1}^{d-2} N_i + N_{i+1} + N_{i+2} \leq k)$ with $N \sim \mathrm{M}_{n,p}$, $n = 500$, $d = 365$, $p = (1/d, \ldots, 1/d)$ and $k \in \{4, \ldots, 25\}$, computed in single-precision.

k	$\underline{p}, \overline{p}$	$\underline{p}, \overline{p}$	e_{abs}	e_{rel}	approx
4	0	0	0	0	0
5	`1.974c00` $\cdot 2^{-135}$ 0	$.0^{40}3652...$ 0	$1.83 \cdot 10^{-41}$	$1.04 \cdot 10^{-2}$	$.0^{40}?$
6	`1.bcc5a4` $\cdot 2^{-67}$ `1.b39300` $\cdot 2^{-67}$	$.0^{19}1177...$ $.0^{19}1152...$	$1.22 \cdot 10^{-22}$	$1.04 \cdot 10^{-2}$	$.0^{19}11?$
7	`1.bbb862` $\cdot 2^{-27}$ `1.b28b40` $\cdot 2^{-27}$	$.0^{7}12913...$ $.0^{7}12646...$	$1.34 \cdot 10^{-10}$	$1.04 \cdot 10^{-2}$	$.0^{7}12?$
8	`1.9d02a2` $\cdot 2^{-11}$ `1.947834` $\cdot 2^{-11}$	$.0^{3}78775...$ $.0^{3}77146...$	$8.15 \cdot 10^{-6}$	$1.04 \cdot 10^{-2}$	$.0^{3}7?$
9	`1.11da84` $\cdot 2^{-4}$ `1.0c30d0` $\cdot 2^{-4}$	$.0668587...$ $.0654762...$	$6.91 \cdot 10^{-4}$	$1.04 \cdot 10^{-2}$	$.06?$
10	`1.867cac` $\cdot 2^{-2}$ `1.7e699a` $\cdot 2^{-2}$	$.3813349...$ $.3734497...$	$3.94 \cdot 10^{-3}$	$1.04 \cdot 10^{-2}$	$.3?$
11	`1.7511fc` $\cdot 2^{-1}$ `1.6d5b2a` $\cdot 2^{-1}$	$.7286528...$ $.7135861...$	$7.53 \cdot 10^{-3}$	$2.7 \cdot 10^{-2}$	$.7?$
12	`1.d331e6` $\cdot 2^{-1}$ `1.c988cc` $\cdot 2^{-1}$	$.9124900...$ $.8936218...$	$9.43 \cdot 10^{-3}$	$9.7 \cdot 10^{-2}$	$.?$
13	`1.f64e04` $\cdot 2^{-1}$ `1.ebeb16` $\cdot 2^{-1}$	$.9810639...$ $.9607779...$	$1.01 \cdot 10^{-2}$	$3.49 \cdot 10^{-1}$	$.9?$
14	1 `1.f6a7a6` $\cdot 2^{-1}$	1 $.9817478...$	$1 - \underline{p}$	∞	$.9?$
15	1 `1.f9a956` $\cdot 2^{-1}$	1 $.9876200...$	$1 - \underline{p}$	∞	$.9?$
16	1 `1.fa6fe6` $\cdot 2^{-1}$	1 $.9891349...$	$1 - \underline{p}$	∞	$.9?$
17	1 `1.fa9fa0` $\cdot 2^{-1}$	1 $.9894990...$	$1 - \underline{p}$	∞	$.9?$
18	1 `1.faaa68` $\cdot 2^{-1}$	1 $.9895813...$	$1 - \underline{p}$	∞	$.9?$
19	1 `1.faacb6` $\cdot 2^{-1}$	1 $.9895989...$	$1 - \underline{p}$	∞	$.9?$
20	1 `1.faad2c` $\cdot 2^{-1}$	1 $.9896024...$	$1 - \underline{p}$	∞	$.9?$
21	1 `1.faad3c` $\cdot 2^{-1}$	1 $.9896029...$	$1 - \underline{p}$	∞	$.9?$
22	1 `1.faad40` $\cdot 2^{-1}$	1 $.9896030...$	$1 - \underline{p}$	∞	$.9?$
23	1 `1.faad44` $\cdot 2^{-1}$	1 $.9896031...$	$1 - \underline{p}$	∞	$.9?$
24	1 `1.faad46` $\cdot 2^{-1}$	1 $.9896032...$	$1 - \underline{p}$	∞	$.9?$
25	1 `1.faad46` $\cdot 2^{-1}$	1 $.9896032...$	$1 - \underline{p}$	∞	$.9?$

line for $k = 15$ in Table 4.5 on page 87, we get the approximate value $1-0.9979961 = 0.0020039$ with an absolute error less than 10^{-7}. As this probability is so small we would, if the described event occurs, reject the hypothesis that the patients arrived independently and hence suspect that there must be a reason for this cluster.

4.6 Computation of rigorous bounds for rectangle scan probabilities for a multivariate hypergeometrically distributed random variable

As stated in 1.3, multivariate hypergeometrically distributed random variables are Markov increments. Therefore implementations of the Algorithm A from Chapter 1 can be used to compute rectangle scan probabilities for a multivariate hypergeometrically distributed random variable. From [6] we take the following example, where we compute rigorous bounds for the exact probabilities.

We use the following algorithm to compute the multivariate hypergeometric transition probabilities, which are univariate hypergeometric. In the "rounding up" mode this algorithm calculates an upper bound for the exact hypergeometric probability. In the "rounding down" mode this algorithm calculates a lower bound for the exact hypergeometric probability.

```
double hyp(int n, int r, int b, int k){
double f=1.0;
int j0=0,j1=0,j2=0;
while ( (j0<k)| (j1<n-k) | (j2<n) ){
if(f<1 && ( (j0<k) | (j1<n-k)) ){
if (j0<k) { f*=(double)(r-j0)/(j0+1);j0++;}
else {if (j1<n-k) { f*=(double)(b-j1)/(j1+1);j1++;}
else if (j2<n) {f*=(double)(r+b-j2)/(j2+1);j2++;}}
}
else if (j2<n) { f*=(double)(j2+1)/(r+b-j2);j2++;}
}
return f;
}
```

Table 4.7 contains the distribution function of the random variable $\max_{i=1}^{d-2}(N_i + N_{i+1} + N_{i+2})$ with $N \sim \mathrm{H}_{n,m}$ with $n = 500$, $d = 365$ and $m = (10, \ldots, 10)$. Details on the used notation are described in 4.5.1. The algorithm with which the values were computed is printed in Appendix F.

Table 4.7: Upper and lower bounds $\overline{p}, \underline{p}$ for $\mathbb{P}(\max_{i=1}^{d-2} N_i + N_{i+1} + N_{i+2} \leq k)$ with $N \sim \mathrm{H}_{n,m}$, $n = 500$, $d = 365$, $m = (10, \ldots, 10)$ and $k \in \{4, \ldots, 26\}$.

k	$[\underline{p}, \overline{p}]$	e_{abs}	e_{rel}	approx
4	0	0	0	**0**
5	`1.94a78cce6bf78` $\cdot 2^{-160}$ `1.94a78cce088a0` $\cdot 2^{-160}$	$3.09 \cdot 10^{-59}$	$2.86 \cdot 10^{-11}$	$.\mathbf{0^{47}10815}$
6	`1.0acc3dae78827` $\cdot 2^{-55}$ `1.0acc3dae36d0e` $\cdot 2^{-55}$	$8.29 \cdot 10^{-28}$	$2.87 \cdot 10^{-11}$	$.\mathbf{0^{16}28926}$
7	`1.591d6928456d6` $\cdot 2^{-20}$ `1.591d6927f05d0` $\cdot 2^{-20}$	$3.69 \cdot 10^{-17}$	$2.87 \cdot 10^{-11}$	$.\mathbf{0^{5}12856}$
8	`1.40ac4ad3593a9` $\cdot 2^{-7}$ `1.40ac4ad30a26f` $\cdot 2^{-7}$	$2.81 \cdot 10^{-13}$	$2.87 \cdot 10^{-11}$	$.\mathbf{0097862}$
9	`1.df885f4b6ceae` $\cdot 2^{-3}$ `1.df885f4af6a55` $\cdot 2^{-3}$	$1.91 \cdot 10^{-11}$	$2.87 \cdot 10^{-11}$	$.\mathbf{2341468}$
10	`1.546bd869a7f5e` $\cdot 2^{-1}$ `1.546bd86953fe9` $\cdot 2^{-1}$	$2.60 \cdot 10^{-11}$	$5.70 \cdot 10^{-11}$	$.\mathbf{6648853}$
11	`1.cec1ebd5b5793` $\cdot 2^{-1}$ `1.cec1ebd543545` $\cdot 2^{-1}$	$2.81 \cdot 10^{-11}$	$2.70 \cdot 10^{-10}$	$.\mathbf{9038233}$
12	`1.f4e8088a29393` $\cdot 2^{-1}$ `1.f4e80889adab5` $\cdot 2^{-1}$	$2.86 \cdot 10^{-11}$	$1.30 \cdot 10^{-9}$	$.\mathbf{9783328}$
13	`1.fde26f4234a4c` $\cdot 2^{-1}$ `1.fde26f41b6dfc` $\cdot 2^{-1}$	$2.87 \cdot 10^{-11}$	$6.92 \cdot 10^{-9}$	$.\mathbf{9958682}$
14	`1.ffa6780ca228e` $\cdot 2^{-1}$ `1.ffa6780c23f48` $\cdot 2^{-1}$	$2.87 \cdot 10^{-11}$	$4.20 \cdot 10^{-8}$	$.\mathbf{9^{3}31693}$
15	`1.fff314a41d498` $\cdot 2^{-1}$ `1.fff314a39f023` $\cdot 2^{-1}$	$2.87 \cdot 10^{-11}$	$2.91 \cdot 10^{-7}$	$.\mathbf{9^{4}01433}$
16	`1.fffe5ec7c001c` $\cdot 2^{-1}$ `1.fffe5ec741b7c` $\cdot 2^{-1}$	$2.87 \cdot 10^{-11}$	$2.31 \cdot 10^{-6}$	$.\mathbf{9^{4}87566}$
17	`1.ffffd2049693f` $\cdot 2^{-1}$ `1.ffffd20418497` $\cdot 2^{-1}$	$2.87 \cdot 10^{-11}$	$2.10 \cdot 10^{-5}$	$.\mathbf{9^{5}8629}?$
18	`1.fffffb9535338` $\cdot 2^{-1}$ `1.fffffb94b6e8e` $\cdot 2^{-1}$	$2.87 \cdot 10^{-11}$	$2.18 \cdot 10^{-4}$	$.\mathbf{9^{6}868}?$
19	`1.fffffa1dc0a3` $\cdot 2^{-1}$ `1.fffffa15dbfb` $\cdot 2^{-1}$	$2.87 \cdot 10^{-11}$	$2.61 \cdot 10^{-3}$	$.\mathbf{9^{7}8}?$
20	`1.fffffff9717b1` $\cdot 2^{-1}$ `1.fffffff8f330a` $\cdot 2^{-1}$	$2.87 \cdot 10^{-11}$	$3.63 \cdot 10^{-2}$	$.\mathbf{9^{9}}?$
21	`1.fffffffffd3bf1` $\cdot 2^{-1}$ `1.fffffffff55749` $\cdot 2^{-1}$	$2.87 \cdot 10^{-11}$	$5.88 \cdot 10^{-1}$	$.\mathbf{9^{10}}?$
22	1 `1.fffffffffbb782` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$.\mathbf{9^{10}}?$
23	1 `1.fffffffffc0de3` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$.\mathbf{9^{10}}?$
24	1 `1.fffffffffc11b4` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$.\mathbf{9^{10}}?$
25	1 `1.fffffffffc11d9` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$.\mathbf{9^{10}}?$
26	1 `1.fffffffffc11d9` $\cdot 2^{-1}$	$1 - \underline{p}$	∞	$.\mathbf{9^{10}}?$

4.7 Calling C-functions from R and changing the rounding mode in R

In R there exists the option to call C-functions. In order to do this, with the C-Compiler first a shared library has to be created that contains the functions which shall be called in R. The shared library, which is located in a file with the file extension .so, then can be included into R with the dyn.load command. After having included the shared library in R, functions can be called from that library using the interface function which is called .C.

By including C-functions it is possible to change the rounding mode in R for example in a way that all the floating point operations $+, -, *, /$ always give results which are rounded downwards and therefore are lower bounds for the exact result.

For example, to functions that change the rounding mode in R, the following C-functions can to be compiled into a shared library with the .so extension.

```
void rounddown(void){
fesetround(FE_DOWNWARD);
}

void roundup(void){
fesetround(FE_UPWARD);
}
```

If the shared library is called RoundingModes.so, then this shared library can be included using the following command in R:

```
dyn.load("RoundingModes.so")
```

After the shared library is included in R, the rounding modes can be changed in R using the following commands

```
.C("rounddown")
```

or

```
.C("roundup")
```

Further information on including C code in R can be found in “Writing R Extensions” at the R project webpage https://cran.r-project.org/manuals.html

A different reason for using C code in R, besides the use of alternate rounding modes, could be the following. Loops in R are said to be slower than in C.

Appendix A

An algorithm for the multinomial range

The following R script prints the values of the cumulative distribution function of the Range

$$D = \max_{i=1}^{d} N_i - \min_{i=1}^{d} N_i$$

for a multinomially distributed random variable $D \sim \mathrm{M}_{n,p}$ with $n = 1000, d = 6$ and $p = (1/d, \ldots, 1/d)$. These values are listed in Table 1.1.

```
rm(list = ls(all = TRUE))

n=1000;
d=6;
p=array(1/d,d)

startprob<- function(i){dbinom(i,n,p[1]) }

markovTransition<- function(k,i,j){
prob=numeric(length(i));
for ( l in 1:length(i)) { prob[l]=dbinom(j-i[l],n-i[l],p[k]/sum(p[k:d])) };
prob
}

multiRectangleProb<- function(n,b,c){

alpha=numeric(d-1)
beta=numeric(d-1)
for (k in 1:(d-1)) alpha[k]=max(n-sum(c[(k+1):d]),sum(b[1:k]))
for (k in 1:(d-1)) beta[k]=min(n-sum(b[(k+1):d]),sum(c[1:k]))

P=numeric(n+1)
```

```
for (j in alpha[1]:beta[1]) P[j+1]=startprob(j)

for (k in 2:(d-1)) {

  Q=numeric(n+1)

    for (x in alpha[k]:beta[k]) {

    su=max(x-c[k],alpha[k-1])
    so=min(x-b[k],beta[k-1])

    if(su<=so) Q[x+1]=sum(markovTransition(k,su:so,x)*P[su:so+1])

    }
  P=Q

}
sum(P)
}


CDFMultiRange<- function(k){
x=0
for (h in 0:(n-k)) {x=x+multiRectangleProb(n,array(h,d),array(h+k,d))}
for (h in 0:(n-k-1)) { x=x-multiRectangleProb(n,array(h+1,d),array(h+k,d))}
x
}


for (k in 1:68) {print(k); print(CDFMultiRange(k))}
```

Appendix B

An algorithm for the cumulative distribution function of a scan statistic of a multinomially distributed random variable

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <fenv.h>


#define max( a, b ) ( ((a) > (b)) ? (a) : (b) )
#define min( a, b ) ( ((a) < (b)) ? (a) : (b) )

void sum(int n,int s, double* startadress, double* sum){
int i;
*sum=0;
for (i=0; i<n; i++){ *sum=*sum + *(startadress+i*s);}
}


double bnp(int k, int n, double p, double q){

if (2*k>n) return(bnp(n-k,n,q,p));

double f=1.0;
int j0=0,j1=0,j2=0;
while ( (j0<k) | (j1<k)| (j2<n-k) )
{
if( (j0<k) && (f<1) ) {
```

```
j0++;
f*= (double)(n-k+j0)/(double)j0;}
else {
if(j1<k) {j1++; f*= p;}
else {j2++; f*= q;}
}
}
return f;
}



void ComputeTransitionProbs(int d, double* pu, double* po,
 double* psu, double* pso, double* qsu, double* qso){

double* sumu = (double*)malloc(8*d);
double* sumo = (double*)malloc(8*d);

sumu[d-1]=pu[d-1];
sumo[d-1]=po[d-1];
fesetround(FE_DOWNWARD);
int i;
for (i=d-1;i>1;i--){
sumu[i-1]= sumu[i]+pu[i-1];
}
fesetround(FE_UPWARD);
for (i=d-1;i>1;i--){
sumo[i-1]= sumo[i]+po[i-1];
}


psu[d-1]=1;
pso[d-1]=1;
psu[0]=pu[0];
pso[0]=po[0];
fesetround(FE_DOWNWARD);
for (i=1;i<d-1;i++){
psu[i]=pu[i]/sumo[i];
if(psu[i]>1){psu[i]=1;}
}
fesetround(FE_UPWARD);
for (i=1;i<d-1;i++){
pso[i]= po[i]/sumu[i];
```

```
if(pso[i]>1){pso[i]=1;}
}


fesetround(FE_DOWNWARD);
for (i=0;i<d;i++){
qsu[i]=1-pso[i];
}
fesetround(FE_UPWARD);
for (i=0;i<d;i++){
qso[i]=1-psu[i];
}


free(sumu);
free(sumo);

return;
}


void Mult3ScanRectangleProb(int d,int l, int n, double* ps,
 double* qs, int* b, int* c, int* m, int* M){
int nn=(n+1)*(n+1);


//Initialize memory
double* P=(double*)malloc(8*(n+1)*(n+1)*(n+1));
double* R=(double*)malloc(8*(n+1)*(n+1)*(n+1));
double* Q;

int i,j,k;
int index;
for(i=0;i<=n;i++){
for(j=i;j<=n;j++){
for(k=j;k<=n;k++){
index=i*nn+j*(n+1)+k;
*(P+index)=0;
*(R+index)=0;
}}}
```

```
//Compute starting probabilities
int ma= min(M[0],n);
int mi= min(m[0],n);

for(i=mi;i<=ma;i++){
for(j=i;j<=ma;j++){
for(k=j;k<=ma;k++){
*(P+i*nn+j*(n+1)+k)
=bnp(i,n,ps[0],qs[0])*bnp(j-i,n-i,ps[1],qs[1])*bnp(k-j,n-j,ps[2],qs[2]);
}}}


//Use recursion to fill the array of probabilities
int nu;
int su,so;

for (nu=2;nu<=d-l+1;nu++){
Q=R;
ma=min(M[nu-1],n);
mi=min(m[nu-1],n);

for(i=mi;i<=ma;i++){
for(j=i;j<=ma;j++){
for(k=j;k<=ma;k++){
su=max(k-c[nu-1],m[nu-2]);
so=min(k-b[nu-1],min(i,M[nu-2]));
index=i*nn+j*(n+1)+k;

if(j<= M[nu-2] && su<= so){
sum(so-su+1,nn,P+su*nn+i*(n+1)+j,Q+index);
*(Q+index)*=bnp(k-j,n-j,ps[nu+1],qs[nu+1]);
}else{*(Q+index)=0;}
}}}

for(i=0;i<=n;i++){
for(j=i;j<=n;j++){
for(k=j;k<=n;k++){
*(P+i*nn+j*(n+1)+k)=0;
}}}

R=P;
P=Q;
}
```

```
//Sum up the relevant entries of the last row of the array
//of probabilities. This yields the result.
double result=0;
ma= min(M[d-l],n);
mi= min(m[d-l],n);
for( i= mi;i<= ma;i++){
for(j=i;j<=ma;j++){
for(k=j;k<=ma;k++){result =  result+ *(P+i*nn+j*(n+1)+k);
}}}
printf("%p ",result);printf("%.20f\n",result);

free(P);
free(R);
return;
}

void Mult3ScanRectangleWrapper(int* D,int* L, int* N,
 double* pu, double* po, int* b, int* c, int* m, int* M){
int d=*D;
int l=*L;
int n=*N;
int nn=(n+1)*(n+1);


//Compute the transition probabilities
double* psu = (double*)malloc(8*d);
double* pso = (double*)malloc(8*d);
double* qsu = (double*)malloc(8*d);
double* qso = (double*)malloc(8*d);
ComputeTransitionProbs(d,pu,po,psu,pso,qsu,qso);


//Compute lower and upper bound for the exact rectangle scan probability
fesetround(FE_DOWNWARD);
Mult3ScanRectangleProb(d,l,n,psu,qsu,b,c,m,M);
fesetround(FE_UPWARD);
Mult3ScanRectangleProb(d,l,n,pso,qso,b,c,m,M);
}

void computation(int d, int l, int n, int k){
double* pu = (double*)malloc(8*d);
double* po = (double*)malloc(8*d);
```

```
int i,j;
for (i=0;i<d;i++){
fesetround(FE_DOWNWARD);
pu[i]=1/(double)d;
fesetround(FE_UPWARD);
po[i]=1/(double)d;
}

int* b=(int*)malloc(4*(d-2));
int* c=(int*)malloc(4*(d-2));
int* m=(int*)malloc(4*(d-2));
int* M=(int*)malloc(4*(d-2));

for(i=0;i<d-l+1;i++){b[i]=0;}
for(i=0;i<d-l+1;i++){c[i]=k;}
for(i=0;i<d-l+1;i++){m[i]=0;}
M[0]=k;
for(i=0;i<d/l-1;i++){
for(j=1;j<=l;j++){M[i*l+j]=k*(i+2);}
}
for(j=1;j<=d-l*(d/l);j++){M[d-l+1-j]=k*(d/l+1);}

double zeit=clock();
Mult3ScanRectangleWrapper(&d,&l, &n,  pu, po, b, c, m, M);
printf("%.2f ",(clock()-zeit)/CLOCKS_PER_SEC);printf("%c\n",'s');
return;
}



int main (void){
int d=365;
int l=3;
int n=500;
int k;

for(k=8;k<=15;k++){
printf("%i\n",k);
computation (d,l,n,k);
}

return 0;
}
```

Appendix C

Stirling's Series

Definition C.1. We define the **Bernoulli Numbers** $b_1, b_2, b_3, \ldots \in \mathbb{R}$ in the following way as the coefficients in the series

$$\frac{x}{\mathrm{e}^x - 1} = 1 - \frac{x}{2} + b_1 \frac{x^2}{2!} - b_2 \frac{x^4}{4!} + \ldots$$

which according to [8] converges for every $x \in \mathbb{R}$ with $|x| < 2\pi$.

Definition C.2. We define **Stirling's Series** $(S_n)_{n\in\mathbb{N}}$ by

$$S_n(x) := \sum_{k=1}^{n} \frac{(-)^{k-1} b_k}{2k(2k-1)} \frac{1}{x^{2k-1}}$$

for $x \in]0, \infty[$ and $n \in \mathbb{N}$.

Example C.3. We have $S_6(x) = \frac{1}{12x} - \frac{1}{360x^3} + \frac{1}{1260x^5} - \frac{1}{1680x^7} + \frac{1}{1188x^9} - \frac{691}{360360x^{11}}$ for $x \in]0, \infty[$.

Let $\mu :]0, \infty[\to \mathbb{R}$

$$\mu(x) := \log\left(\frac{\Gamma(x+1)}{x^{x+\frac{1}{2}}\mathrm{e}^{-x}\sqrt{2\pi}}\right)$$

where $\Gamma :]0, \infty[\to \mathbb{R}$ is the **Gamma-Function** which is defined by the conditions

$$\begin{aligned}
&\Gamma(x+1) = x \cdot \Gamma(x) \text{ for } x \in]0, \infty[\\
&\Gamma(1) = 1 \\
&\log(\Gamma) \text{ is convex}
\end{aligned}$$

From [18] we know the following theorem which states approximations of μ by Stirling's series. The theorem is proven with the help of results from [23].

Theorem C.4. *For $x \to \infty$ we have the asymptotic expansion*

$$\mu(x) \quad \sim \quad (S_n(x))_{n\in\mathbb{N}}$$

That means, that for every $n \in \mathbb{N}$ we have

$$\mu(x) - S_n(x) = o(S_n(x) - S_{n-1}(x)) \text{ for } x \to \infty$$

Moreover, for every $x \in]0, \infty[$ the series $(S_n(x))_{n\in\mathbb{N}}$ is enveloping the value $\mu(x)$, which means that we have the inequalities

$$S_{n-1}(x) \geq \mu(x) \geq S_n(x)$$

for every $n \in \mathbb{N}$ which is even.

Example C.5. We have

$$S(x) \leq \frac{1}{12x} \tag{C.1}$$

and

$$S(x) \geq \frac{1}{12x} - \frac{1}{360x^3} \tag{C.2}$$

and therefore

$$\left| S(x) - \frac{1}{12x} \right| \leq \frac{1}{360x^3}$$

for every $x \in]0, \infty[$, and therefore the relative error

$$\frac{|S(x) - \frac{1}{12x}|}{|S(x)|} \leq \frac{1}{360x^3(\frac{1}{12x} - \frac{1}{360x^3})} = \frac{1}{30x^2 - 1}$$

for every $x \in]\frac{1}{\sqrt{30}}, \infty[$.

Appendix D

Loader's algorithm for the binomial density

This is a print of the files dbinom.c, bd0.c and stirlerr.c, called from the folder http://svn.r-project.org/R/trunk/src/nmath/ on October 9th 2014. These files contain the program code for the function `dbinom`, that the R version 3.1.0 (2014-04-10) uses to compute the binomial density. Uwe Ligges [15] wrote an article that describes how the C-Code of every built-in R-function can be displayed. We assume that the C code is written according to the C Standard [2]. This algorithm for the binomial density is a slightly modified version of the algorithm described in [16].

D.1 Print of the file dbinom.c

```
/*
 * AUTHOR
 *   Catherine Loader, catherine@research.bell-labs.com.
 *   October 23, 2000.
 *
 *  Merge in to R and further tweaks :
 * Copyright (C) 2000-2014 The R Core Team
 * Copyright (C) 2008 The R Foundation
 *
 *  This program is free software; you can redistribute it and/or modify
 *  it under the terms of the GNU General Public License as published by
 *  the Free Software Foundation; either version 2 of the License, or
 *  (at your option) any later version.
 *
 *  This program is distributed in the hope that it will be useful,
 *  but WITHOUT ANY WARRANTY; without even the implied warranty of
```

```
 *  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *  GNU General Public License for more details.
 *
 *  You should have received a copy of the GNU General Public License
 *  along with this program; if not, a copy is available at
 *  http://www.r-project.org/Licenses/
 *
 *
 * DESCRIPTION
 *
 *   To compute the binomial probability, call dbinom(x,n,p).
 *   This checks for argument validity, and calls dbinom_raw().
 *
 *   dbinom_raw() does the actual computation; note this is called by
 *   other functions in addition to dbinom().
 *     (1) dbinom_raw() has both p and q arguments, when one may be represented
 *         more accurately than the other (in particular, in df()).
 *     (2) dbinom_raw() does NOT check that inputs x and n are integers. This
 *         should be done in the calling function, where necessary.
 *         -- but is not the case at all when called e.g., from df() or dbeta() !
 *     (3) Also does not check for 0 <= p <= 1 and 0 <= q <= 1 or NaN's.
 *         Do this in the calling function.
 */

#include "nmath.h"
#include "dpq.h"

double attribute_hidden
dbinom_raw(double x, double n, double p, double q, int give_log)
{
    double lf, lc;

    if (p == 0) return((x == 0) ? R_D__1 : R_D__0);
    if (q == 0) return((x == n) ? R_D__1 : R_D__0);

    if (x == 0) {
if(n == 0) return R_D__1;
lc = (p < 0.1) ? -bd0(n,n*q) - n*p : n*log(q);
return( R_D_exp(lc) );
    }
    if (x == n) {
lc = (q < 0.1) ? -bd0(n,n*p) - n*q : n*log(p);
return( R_D_exp(lc) );
```

```
    }
    if (x < 0 || x > n) return( R_D__0 );

    /* n*p or n*q can underflow to zero if n and p or q are small.
       This used to occur in dbeta, and gives NaN as from R 2.3.0.  */
    lc = stirlerr(n) - stirlerr(x) - stirlerr(n-x)
           - bd0(x,n*p) - bd0(n-x,n*q);

    /* f = (M_2PI*x*(n-x))/n; could overflow or underflow */
    /* Upto R 2.7.1:
     * lf = log(M_2PI) + log(x) + log(n-x) - log(n);
     * -- following is much better for  x << n : */
    lf = M_LN_2PI + log(x) + log1p(- x/n);

    return R_D_exp(lc - 0.5*lf);
}

double dbinom(double x, double n, double p, int give_log)
{
#ifdef IEEE_754
    /* NaNs propagated correctly */
    if (ISNAN(x) || ISNAN(n) || ISNAN(p)) return x + n + p;
#endif

    if (p < 0 || p > 1 || R_D_negInonint(n))
ML_ERR_return_NAN;
    R_D_nonint_check(x);
    if (x < 0 || !R_FINITE(x)) return R_D__0;

    n = R_forceint(n);
    x = R_forceint(x);

    return dbinom_raw(x, n, p, 1-p, give_log);
}
```

D.2 Print of the file bd0.c

```
/*
 *  AUTHOR
 * Catherine Loader, catherine@research.bell-labs.com.
```

```
 * October 23, 2000.
 *
 *  Merge in to R:
 * Copyright (C) 2000, The R Core Team
 *
 *  This program is free software; you can redistribute it and/or modify
 *  it under the terms of the GNU General Public License as published by
 *  the Free Software Foundation; either version 2 of the License, or
 *  (at your option) any later version.
 *
 *  This program is distributed in the hope that it will be useful,
 *  but WITHOUT ANY WARRANTY; without even the implied warranty of
 *  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *  GNU General Public License for more details.
 *
 *  You should have received a copy of the GNU General Public License
 *  along with this program; if not, a copy is available at
 *  http://www.r-project.org/Licenses/
 *
 *
 *  DESCRIPTION
 * Evaluates the "deviance part"
 * bd0(x,M) :=  M * D0(x/M) = M*[ x/M * log(x/M) + 1 - (x/M) ] =
 *   =  x * log(x/M) + M - x
 * where M = E[X] = n*p (or = lambda), for   x, M > 0
 *
 * in a manner that should be stable (with small relative error)
 * for all x and M=np. In particular for x/np close to 1, direct
 * evaluation fails, and evaluation is based on the Taylor series
 * of log((1+v)/(1-v)) with v = (x-M)/(x+M) = (x-np)/(x+np).
 */
#include "nmath.h"

double attribute_hidden bd0(double x, double np)
{
    double ej, s, s1, v;
    int j;

    if(!R_FINITE(x) || !R_FINITE(np) || np == 0.0) ML_ERR_return_NAN;

    if (fabs(x-np) < 0.1*(x+np)) {
v = (x-np)/(x+np);  // might underflow to 0
s = (x-np)*v;/* s using v -- change by MM */
```

```
if(fabs(s) < DBL_MIN) return s;
ej = 2*x*v;
v = v*v;
for (j = 1; j < 1000; j++) { /* Taylor series; 1000: no infinite loop
as |v| < .1,  v^2000 is "zero" */
    ej *= v;// = v^(2j+1)
    s1 = s+ej/((j<<1)+1);
    if (s1 == s) /* last term was effectively 0 */
return s1 ;
    s = s1;
}
    }
    /* else:  | x - np |  is not too small */
    return(x*log(x/np)+np-x);
}
```

D.3 Print of the file stirlerr.c

```
/*
 *  AUTHOR
 *    Catherine Loader, catherine@research.bell-labs.com.
 *    October 23, 2000.
 *
 *  Merge in to R:
 * Copyright (C) 2000, The R Core Team
 *
 *  This program is free software; you can redistribute it and/or modify
 *  it under the terms of the GNU General Public License as published by
 *  the Free Software Foundation; either version 2 of the License, or
 *  (at your option) any later version.
 *
 *  This program is distributed in the hope that it will be useful,
 *  but WITHOUT ANY WARRANTY; without even the implied warranty of
 *  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *  GNU General Public License for more details.
 *
 *  You should have received a copy of the GNU General Public License
 *  along with this program; if not, a copy is available at
 *  http://www.r-project.org/Licenses/
 *
```

```
 *
 *  DESCRIPTION
 *
 *    Computes the log of the error term in Stirling's formula.
 *      For n > 15, uses the series 1/12n - 1/360n^3 + ...
 *      For n <=15, integers or half-integers, uses stored values.
 *      For other n < 15, uses lgamma directly (don't use this to
 *        write lgamma!)
 *
 * Merge in to R:
 * Copyright (C) 2000, The R Core Team
 * R has lgammafn, and lgamma is not part of ISO C
 */

#include "nmath.h"

/* stirlerr(n) = log(n!) - log( sqrt(2*pi*n)*(n/e)^n )
 *             = log Gamma(n+1) - 1/2 * [log(2*pi) + log(n)] - n*[log(n) - 1]
 *             = log Gamma(n+1) - (n + 1/2) * log(n) + n - log(2*pi)/2
 *
 * see also lgammacor() in ./lgammacor.c  which computes almost the same!
 */

double attribute_hidden stirlerr(double n)
{

#define S0 0.083333333333333333       /* 1/12 */
#define S1 0.00277777777777777777     /* 1/360 */
#define S2 0.00079365079365079365079365  /* 1/1260 */
#define S3 0.00059523809523809523809523809 /* 1/1680 */
#define S4 0.00084175084175084175084175084175/* 1/1188 */

/*
  error for 0, 0.5, 1.0, 1.5, ..., 14.5, 15.0.
*/
    const static double sferr_halves[31] = {
0.0, /* n=0 - wrong, place holder only */
0.153426409720027345291384 8,  /* 0.5 */
0.081061466795327258219670 2,  /* 1.0 */
0.054814121051917653896139 0,  /* 1.5 */
0.041340695955409294093822 1,  /* 2.0 */
0.033162873519936287485110 48, /* 2.5 */
0.027677925684998339148789 29, /* 3.0 */
```

```
0.023746163656297495971329 20, /* 3.5 */
0.020790672103765093111522 77, /* 4.0 */
0.018488450532673185230779 34, /* 4.5 */
0.016644691189821192163194 87, /* 5.0 */
0.015134973221917378873512 55, /* 5.5 */
0.013876128823070747998745 73, /* 6.0 */
0.012810465242920226924249 86, /* 6.5 */
0.011896709945891770095055 72, /* 7.0 */
0.011104559758206917326629 91, /* 7.5 */
0.010411265261972096497478 567, /* 8.0 */
0.009799416126158803298389 475, /* 8.5 */
0.009255462182712732917728 637, /* 9.0 */
0.008768700134139385462952 823, /* 9.5 */
0.008330563433362871256469 318, /* 10.0 */
0.007934114564314020547248 100, /* 10.5 */
0.007573675487951840794972 024, /* 11.0 */
0.007244554301320383179543 912, /* 11.5 */
0.006942840107209529865664 152, /* 12.0 */
0.006665247032707682442354 394, /* 12.5 */
0.006408994188004207068439 631, /* 13.0 */
0.006171712263039457647532 867, /* 13.5 */
0.005951370112758847735624 416, /* 14.0 */
0.005746216513010115682023 589, /* 14.5 */
0.005554733551962801371038 690  /* 15.0 */
    };
    double nn;

    if (n <= 15.0) {
nn = n + n;
if (nn == (int)nn) return(sferr_halves[(int)nn]);
return(lgammafn(n + 1.) - (n + 0.5)*log(n) + n - M_LN_SQRT_2PI);
    }

    nn = n*n;
    if (n>500) return((S0-S1/nn)/n);
    if (n> 80) return((S0-(S1-S2/nn)/nn)/n);
    if (n> 35) return((S0-(S1-(S2-S3/nn)/nn)/nn)/n);
    /* 15 < n <= 35 : */
    return((S0-(S1-(S2-(S3-S4/nn)/nn)/nn)/nn)/n);
}
```

Appendix E

Computation of the Poisson density

With the help of a rounding error estimate for the functions `stirlerr` and `bd0` also a rounding error estimate for the following algorithm for the computation of the Poisson density

$$\mathrm{p}_\lambda(x) := \frac{\lambda^x}{x!}\mathrm{e}^{-\lambda}$$

can be done. This algorithm is proposed by Loader [16], Appendix A, and is used by the statistical software R to compute the poisson density. It was called from the folder https://svn.r-project.org/R/trunk/src/nmath/ on July 17 2016.

```
#include "nmath.h"
#include "dpq.h"

double attribute_hidden dpois_raw(double x, double lambda, int give_log)
{
    /*       x >= 0 ; integer for dpois(), but not e.g. for pgamma()!
        lambda >= 0
    */
    if (lambda == 0) return( (x == 0) ? R_D__1 : R_D__0 );
    if (!R_FINITE(lambda)) return R_D__0;
    if (x < 0) return( R_D__0 );
    if (x <= lambda * DBL_MIN) return(R_D_exp(-lambda) );
    if (lambda < x * DBL_MIN) return(R_D_exp(-lambda
        + x*log(lambda) -lgammafn(x+1)));
    return(R_D_fexp( M_2PI*x, -stirlerr(x)-bd0(x,lambda) ));
}

double dpois(double x, double lambda, int give_log)
```

```
{
#ifdef IEEE_754
    if(ISNAN(x) || ISNAN(lambda))
        return x + lambda;
#endif

    if (lambda < 0) ML_ERR_return_NAN;
    R_D_nonint_check(x);
    if (x < 0 || !R_FINITE(x))
return R_D__0;

    x = R_forceint(x);

    return( dpois_raw(x,lambda,give_log) );
}
```

Appendix F

An algorithm for the cumulative distribution function of a scan statistic of a multivariate hypergeometrically distributed random variable

The following algorithm computes the probability $\mathbb{P}(\max_{i=1}^{d-2}(N_i + N_{i+1} + N_{i+2}) \leq k)$ for a multivariate hypergeometrically distributed random variable $N \sim \mathrm{H}_{n,m}$ with $n = 500, d = 365, m = (10, \ldots, 10) \in \mathbb{R}^d$ and $k \in \{4, \ldots, 26\}$.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <fenv.h>
#define max( a, b ) ( ((a) > (b)) ? (a) : (b) )
#define min( a, b ) ( ((a) < (b)) ? (a) : (b) )
#define sumw(n,s,startadresse,summe)
*summe=0;for(lindex=0;lindex<n;lindex++)
{*summe+=*(startadresse+lindex*s);}

void sum(int n, double* startadresse, double* sum){
int i;
*sum=0;
for (i=0; i<n; i++){ *sum+=*(startadresse-i);}
}

double hyp(int n, int r, int b, int k){
if( (k<max(0,n-b)) | (k>min(n,r)) | (n>r+b) ) {return 0.0;}
if(b==0){ if(r>=n && k==n){return 1.0;} else {return 0.0;}}
```

```
double f=1.0;
int j0=0,j1=0,j2=0;
while ( (j0<k)| (j1<n-k) | (j2<n) ){
if(f<1 && ( (j0<k) | (j1<n-k)) ){
if (j0<k) { f*=(double)(r-j0)/(j0+1);j0++;}
else {if (j1<n-k) { f*=(double)(b-j1)/(j1+1);j1++;} else if (j2<n)
{f*=(double)(r+b-j2)/(j2+1);j2++;}}
}
else if (j2<n) { f*=(double)(j2+1)/(r+b-j2);j2++;}
}
return f;
}



void computeTransitionProb(int d, int* p, int* q){
q[d-1]=0;
q[d-2]=p[d-1];

int i;
for (i=d-1;i>0;i--){
q[i-1]= q[i]+p[i-1];
}
return;
}



double Hyper3ScanRectangleProb(int D, int L, int N, int* p, int* q,
 int* B, int* C, int* mini, int* maxi){


unsigned int d=(unsigned int)D;
unsigned int l=(unsigned int)L;
unsigned int n=(unsigned int)N;
unsigned int* b=(unsigned int*)B;
unsigned int* c=(unsigned int*)C;
unsigned int* m=(unsigned int*)mini;
unsigned int* M=(unsigned int*)maxi;
unsigned int lindex;

unsigned int n1=n+1;
unsigned int n2=n1*n1;
```

```
size_t w=(size_t)n;
w=sizeof(double)*(w+1)*(w+1)*(w+1);

//Initialize memory
double* P=(double*)malloc(w);
double* R=(double*)malloc(w);
double* Q;

unsigned int i,j,k;
unsigned int index;
for(i=0;i<=n;i++){
for(j=i;j<=n;j++){
for(k=j;k<=n;k++){
index=i*n2+j*n1+k;
*(P+index)=0;
*(R+index)=0;
}}}



//Compute starting probabilities
unsigned int ma= min(M[0],n);
unsigned int mi= m[0];
for(i=mi;i<=ma;i++){
for(j=i;j<=ma;j++){
for(k=j;k<=ma;k++){
*(P+i*n2+j*n1+k)=hyp(n,p[0],q[0],i)*hyp(n-i,p[1],q[1],j-i)
*hyp(n-j,p[2],q[2],k-j);
}}}


//Use recursion to fill the array
unsigned int nu;
unsigned int su,so;
unsigned int jo,ko;
for (nu=2;nu<=d-l+1;nu++){
Q=R;
ma=min(M[nu-1],n);
mi=m[nu-1];

for(i=mi;i<=ma;i++){
jo=min(ma,M[nu-2]);
ko=min(ma,c[nu-1]+i);
```

```
for(j=i;j<=jo;j++){
for(k=j;k<=ko;k++){
su=(c[nu-1]<k)?max(m[nu-2],k-c[nu-1]):m[nu-2];
so=(b[nu-1]<k)?min(k-b[nu-1],min(i,M[nu-2])):min(i,M[nu-2]);
if(su<=so){
index=i*n2+j*n1+k;
sumw(so-su+1,n2,P+su*n2+i*n1+j,(Q+index));
*(Q+index)*= hyp(n-j,p[nu+1],q[nu+1],k-j);
}
}}}
ma=min(M[nu-2],n);
for(i=m[nu-2];i<=ma;i++){
for(j=i;j<=ma;j++){
for(k=j;k<=ma;k++){
*(P+i*n2+j*n1+k)=0;
};}}

R=P;
P=Q;
}
```

```
//Sum up the relevant entries of the last row of the array. This yields the result.
double result=0;
ma= min(M[d-l],n);
mi= m[d-l];
for( i= mi;i<= ma;i++){
for(j=i;j<=ma;j++){
for(k=j;k<=ma;k++){result +=  *(P+i*n2+j*n1+k);
}}}
```

```
free(P);
free(R);
return result;
}
```

```
int main(void){
double time;
double upperbound, lowerbound;
int n=500;
int d=365;
int l=3;
int k=6;
int* p=malloc(d*sizeof(int));
int* q=malloc(d*sizeof(int));
int* b=malloc(d*sizeof(int));
int* c=malloc(d*sizeof(int));
int i,j;
for(i=0;i<d;i++){
*(p+i)=10;
}


computeTransitionProb(d,p,q);



for(k=4;k<27;k++){

for(i=0;i<d-l+1;i++){
*(b+i)=0;*(c+i)=k;
}

int* mini=(int*)malloc((d-2)*sizeof(int));
int* maxi=(int*)malloc((d-2)*sizeof(int));

for(i=0;i<d-l+1;i++){mini[i]=0;}
maxi[0]=k;
for(i=0;i<d/l-1;i++){
for(j=1;j<=l;j++){maxi[i*l+j]=k*(i+2);}
}
for(j=1;j<=d-l*(d/l);j++){maxi[d-l+1-j]=k*(d/l+1);}

printf("%i\n",k);
fesetround(FE_DOWNWARD);
time=clock();
lowerbound=Hyper3ScanRectangleProb(d,l,n,p,q,b,c,mini,maxi);
printf("%.2f ",(clock()-time)/CLOCKS_PER_SEC);printf("%c\n",'s');
```

```
printf("%p\n",lowerbound);
fesetround(FE_UPWARD);
time=clock();
upperbound=Hyper3ScanRectangleProb(d,l,n,p,q,b,c,mini,maxi);
printf("%.2f ",(clock()-time)/CLOCKS_PER_SEC);printf("%c\n",'s');
printf("%p\n",upperbound);
printf("%.60f\n",lowerbound);
printf("%.60f\n",upperbound);
printf("\n");
}
return 0;
}
```

Appendix G

An enumerative algorithm for multinomial rectangle scan probabilities

```
#include <stdio.h>
#include <time.h>

double bnp(unsigned int n, double p, unsigned int k){
double result=1;
int i;
for(i=1;i<=k;i++){
result=result * (double)(n-k+i)/(double)i * p;
}
for(i=1;i<=n-k;i++){
result= result * (1-p);
}
return result;
}

double mult12(unsigned int n, unsigned int i1, unsigned int i2,
 unsigned int i3, unsigned int i4, unsigned int i5, unsigned int i6,
 unsigned int i7, unsigned int i8, unsigned int i9, unsigned int i10,
 unsigned int i11, unsigned int i12){

return bnp(n,1/(double)12,i1)*bnp(n-i1,1/(double)11,i2)
*bnp(n-i1-i2,1/(double)10,i3)*bnp(n-i1-i2-i3,1/(double)9,i4)
*bnp(n-i1-i2-i3-i4,1/(double)8,i5)*bnp(n-i1-i2-i3-i4-i5,1/(double)7,i6)
*bnp(n-i1-i2-i3-i4-i5-i6,1/(double)6,i7)
*bnp(n-i1-i2-i3-i4-i5-i6-i7,1/(double)5,i8)
*bnp(n-i1-i2-i3-i4-i5-i6-i7-i8,1/(double)4,i9)
*bnp(n-i1-i2-i3-i4-i5-i6-i7-i8-i9,1/(double)3,i10)
```

```
*bnp(n-i1-i2-i3-i4-i5-i6-i7-i8-i9-i10,1/(double)2,i11)
*bnp(n-i1-i2-i3-i4-i5-i6-i7-i8-i9-i10-i11,(double)1,i12);
}

int main (void){

unsigned int n = 20;
unsigned int d = 12;
unsigned int k = 9;
double result = 0;
double time=clock();


unsigned int i,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,i12;
for( i1=0; i1<= k; i1++){
for( i2=0; i2<= k; i2++){
for( i3=0; i3<= k; i3++){
for( i4=0; i4<= k; i4++){
for( i5=0; i5<= k; i5++){
for( i6=0; i6<= k; i6++){
for( i7=0; i7<= k; i7++){
for( i8=0; i8<= k; i8++){
for( i9=0; i9<= k; i9++){
for( i10=0; i10<= k; i10++){
for( i11=0; i11<= k; i11++){
for( i12=0; i12<= k; i12++){
if(i1+i2+i3+i4+i5+i6+i7+i8+i9+i10+i11+i12==n && (i1+i2+i3<=k) && (i2+i3+i4<=k)
&& (i3+i4+i5<=k) && (i4+i5+i6<=k) && (i5+i6+i7<=k) && (i6+i7+i8<=k)
&& (i7+i8+i9<=k) && (i8+i9+i10<=k)&& (i9+i10+i11<=k) && (i10+i11+i12<=k)){
result=result+mult12(n,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,i12);
}
}}}}}}}}}}}}
printf("%.2f ",(clock()-time)/CLOCKS_PER_SEC);printf("%c\n",'s');
printf("%f\n",result);

return 0;
}
```

Bibliography

[1] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1964.

[2] ANSI. ISO/IEC 9899: Programming languages - C. 1999.

[3] ANSI/IEEE. Standard 754-1985 for binary floating-point arithmetic (also IEC 60559). 1985.

[4] C.J. Corrado. The exact distribution of the maximum, minimum and the range of multinomial/dirichlet and multivariate hypergeometric frequencies. *Statistics and Computing*, 21:349–359, 2011.

[5] J. Dimitriadis. *Die Verteilungsfunktion des Multinomial-Maximums. Algorithmen und Approximationen*. Universität zu Lübeck, 2009. Diploma Thesis.

[6] J. Dimitriadis. Rigorous computing of rectangle scan probabilities for markov increments. *Preprint, arXiv:1109.3254*, 2011.

[7] Norbert Henze. *Stochastik für Einsteiger - Eine Einführung in die faszinierende Welt des Zufalls*. Springer-Verlag, Berlin Heidelberg New York, 2013.

[8] Harro Heuser. *Lehrbuch der Analysis 1*. Teubner, Stuttgart, 1980.

[9] N. J. Higham. *Accuracy and Stability of Numerical Algorithms - Second Edition*. SIAM, Philadelphia, 2nd edition, 2002.

[10] Y. Hirai and T. Nakamura. A new arithmetic and an application to the computation of binomial probability for very wide range of sample size. *Japanese J. Appl. Statisti.*, 35 (2):93–111, 2006.

[11] Norman Lloyd Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Discrete multivariate distributions*, volume 165. Wiley New York, 1997.

[12] R. Kaiser. Binomial probabilities. *Online: https://www.soa.org/News-and-Publications/Newsletters/Compact/2015/march/Binomial-Probabilities.aspx*, 2015.

[13] Donald E. Knuth. *The Art of Computer Programming, Volume 2 - Seminumerical Algorithms*. Addison-Wesley Professional, Boston, 3rd edition, 2014.

[14] U. Kulisch. *Computer Arithmetic and Validity - Theory, Implementation, and Applications*. Walter de Gruyter, Berlin, 2013.

[15] Uwe Ligges. R Help Desk: Accessing the sources. *R News*, 6(4):43–45, October 2006.

[16] C. Loader. Fast and accurate computation of binomial probabilities. *Online: https://lists.gnu.org/archive/html/octave-maintainers/2011-09/pdfK0uKOST642.pdf, accessed 11 July 2016*, 2000.

[17] D. Pfeifer. Strichlisten bei Laplace-Experimenten - zum Paradox der ungleichmäßigen Verteilung. *Stochastik in der Schule*, 26:23–27, 2006.

[18] George Pólya and Gabor Szegö. *Problems and Theorems in Analysis I: Series. Integral Calculus. Theory of Functions*. Springer Science & Business Media, 1998.

[19] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical recipes in C*, volume 2. Cambridge university press Cambridge, 1996.

[20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

[21] Walter Rudin. *Principles of mathematical analysis*. McGraw-Hill Book Co., New York, third edition, 1976. International Series in Pure and Applied Mathematics.

[22] William T Vetterling, Saul A Teukolsky, William H Press, and Brian P Flannery. *Numerical recipes example book (C)*. JSTOR, 1992.

[23] Edmund Taylor Whittaker and George Neville Watson. *A course of modern analysis*. Cambridge university press, 1996.