Peter Zinterhof, Peter Schmitzberger

University of Salzburg
Research Institute for Software Technology
RIST++

Integrate: A Kernel of FORTRAN and C package PANNUM for High Dimensional Number Theoretic Quadrature

1 Motivation

Number Theoretic Methods are a most promising way in contemporary High Dimensional Numerics and in High Performance Scientific Computing. Number Theoretic Numerics (NTN) has been developped from the first deep results in the sixties (E. Hlawka, N. M. Korobov) to a well established branch of Applicable Analysis and Numerics. Considerable progress has been made in the last years by austrian mathematicians especially in the field of best possible methods and error estimations. The probably best and topical reference is Niederreiter [4]. On the other hand few implementations especially on parallel computers are known.

The present work is part of the Salzburg New Technology of Numerics Project (NEWTON). The goal of NEWTON is the development of NTN in the directon of practicability. In the first steps the Salzburg group will do mathematical research in the field of parallelization and implementation of algorithms for high dimensional numerical integration (smooth and non smooth integrands), Integral Equations, Random Vectors and others.

2 Introduction

We concentrate in this paper on number theoretic integration methods. Number theoretic methods are preferable applied to high dimensional problems. The essence of the methods is probably most easily seen considering Korobov's [3] original Fourier Series approach:

For the sake of simplicity lets assume an absolute convergent Fourier Series $f(x_1, \ldots, x_s)$ on the s-dimensional unit cube $I^s = [0,1)^s$:

$$f(x_1, \ldots, x_s) = \sum_{m_1, \ldots, m_s = -\infty}^{\infty} c(m_1, \ldots, m_s) \exp(2\pi i(m_1 x_1 + \cdots + m_s x_s)) \quad (1)$$

For the Fourier coefficients of f holds

$$c(m_1, \ldots, m_s) = \int_{I^s} f(x_1, \ldots, x_s) \exp(-2\pi i(m_1 x_1 + \cdots + m_s x_s))\, dx_1 \cdots dx_s \quad (2)$$

especially

$$c(0, \ldots, 0) = \int_{I^s} f(x_1, \ldots, x_s)\, dx_1 \cdots dx_s \quad (3)$$

Now let $(x_1^k, \ldots, x_s^k), k = 1, \ldots, N$, be a set of nodes for the integration of f by the method

$$R_N(f) = \frac{1}{N} \sum_{k=1}^{N} f(x_1^k, \ldots, x_s^k) - \int_{I^s} f(x_1, \ldots, x_s)\, dx_1 \cdots dx_s. \quad (4)$$

Using the Fourier expansion of $f(x_1, \ldots, x_s)$ one gets immediately the explicit representation of the error term $R_N(f)$ by means of an infinite series

$$R_N(f) = \sum_{m_1, \ldots, m_s \neq 0, \ldots, 0} c(m_1, \ldots, m_s) \frac{1}{N} \sum_{k=1}^{N} \exp(2\pi i(m_1 x_1^k + \cdots + m_s x_s^k)) \quad (5)$$

So it becomes clear: The error term becomes small if the trigonometric sums

$$\frac{1}{N} \sum_{k=1}^{N} \exp(2\pi i(m_1 x_1^k + \cdots + m_s x_s^k)) \quad (6)$$

are (frequently) small and if the Fourier coefficients of f vanish rapidly at infinity, i. e. if f has a smooth behavior with respect to the Fourier orthonormal system. The study of trigonometric sums is an important topic in Number Theory. So one can use a lot of number theoretical results to achieve good integration rules.

We concentrate in this paper on four very different methods of construction of integration rules, each of them having some advantages of own.

Because the assumption of absolute convergence of the Fourier Series does not admit concrete error estimates we introduce according to Korobov [3] the classes $E_s^\alpha(C)$ iff

$$|c(m_1, \ldots, m_s)| \leq C|(\overline{m}_1\overline{m}_2 \cdots \overline{m}_s)^{-\alpha}|, \tag{7}$$

where $\overline{m} = \max(1, |m|)$.

Let $\{x\}$ be the fractional part of the real x, $\{x\} = x - [x]$, $[x]$ being the integer part of x. In other words $\{x\} \equiv x \bmod 1$, $0 \leq \{x\} < 1$.

Now we describe the four different methods of constructing integration rules and their numerical behaviour.

2.1 Integration rules

2.1.1 k-Theta-sequences

Let $(\Theta_1, \ldots, \Theta_s)$ be a real vector. Then

$$(x_1^k, x_2^k, \ldots, x_s^k) = (\{k\Theta_1\}, \ldots, \{k\Theta_s\}) \qquad k = 1, 2, 3, \ldots \tag{8}$$

is called a k-Theta-sequence. Obviously this sequence $\underline{x}^k = (x_1^k, \ldots, x_s^k)$ is periodic iff $(\Theta_1, \ldots, \Theta_s)$ is rational. The $\underline{x}^k, k = 1, 2, \ldots, N, \ldots$ form an infinite semi group modulo 1. The infinite sequence $\underline{x}^k, k = 1, 2, \ldots, N, \ldots$ is uniformly distributed mod 1 iff $(\Theta_1, \ldots, \Theta_s)$ is irrational. In this case the sequence $\underline{x}^k, k = 1, 2, \ldots, N, \ldots$ can be used as infinite set of integration rules which converges at least for all continous functions. This method has an important property: Because the $\underline{x}^k$ don't depend on N and because

$$R_N = \frac{1}{N} \sum_{k=1}^{N} f(\underline{x}^k) - \int_{I^s} f(\underline{x})\, d\underline{x} \tag{9}$$

the incrementation $N \rightarrow N + 1$ does not cause any loss of calculations previous done. This fact could be regarded as the price of a rather slow convergence of $R_N(f) \rightarrow 0$. Zinterhof [6] showed that for almost all $(\Theta_1, \ldots, \Theta_s)$ in the sense of Lebesgue-measure on the classes $E_s^\alpha(C), \alpha > 1$ the best possible estimation

$$R_N = O\left(\frac{1}{N}\right) \tag{10}$$

holds and that this estimation can be actually reached using the irrationals $\Theta_1 = \exp(r_1), \ldots, \Theta_s = \exp(r_s), r_1, \ldots, r_s$ being different nonzero rationals. A much better error estimation can be proven using Bakhvalov's weights $\rho_{k,M}^t$ which are simply defined by

$$\left(\sum_{k=0}^{M} z^k\right)^t = \sum_{k=0}^{Mt} \rho_{k+1,M}^t z^k \tag{11}$$

Using the rule ($t \geq \alpha > 1\,, N = Mt+1$)

$$R_N(f) = \frac{1}{(M+1)^t} \sum_{k=1}^{Mt+1} \rho_{k,M}^t f(\underline{x}^k) - \int_{I^s} f(\underline{x})\, d\underline{x} \tag{12}$$

one gets the error estimation

$$R_N(f) = O\left(\frac{1}{N^{\alpha-\epsilon}}\right)\,, \epsilon > 0 \tag{13}$$

for the nodes $\underline{x}^k$ as before.

Despite of the fact that the proofs of these estimates are by no means trivial the methods are easily implemented. Furthermore the pseudo random vectors $\underline{x}^k$, $k = 1, 2, \ldots, N, \ldots$ have rather good distribution properties, the Kolmogorov distance D_N between the empirical distribution function of the $\underline{x}^1, \ldots, \underline{x}^N$ and the uniform distribution function is of the order

$$D_N = O\left(\frac{1}{N^{1-\epsilon}}\right)\,, \epsilon > 0 \tag{14}$$

whereas $D_N = O(1/N)$ being the best possible estimation which latter is only achieved in the trivial onedimensional case.

2.1.2 Parallelepipedic lattices

Let $a_1, \ldots, a_s$ be integers. Then

$$\underline{x}^k = (\left\{\frac{a_1 k}{N}\right\}, \ldots, \left\{\frac{a_s k}{N}\right\}) \qquad k = 1, \ldots, N \tag{15}$$

is called a parallelepipedic lattice. Parallelepipedic lattices form a group mod N, the sets $\left\{\underline{x}^k(N), k = 1, \ldots, N\right\}$ are changing completely if N changes, such that by changing N all the computational work previously done will be lost. The group structure of the sets of nodes $\underline{x}^k = (\left\{\frac{a_1 k}{N}\right\}, \ldots, \left\{\frac{a_s k}{N}\right\}), k = 1, \ldots, N$ is frequently useful (e. g. considering Integral equations of the convolution type). Korobov introduced parallelepipedic lattices and proved for the special choice of $a_1, \ldots, a_s$ as so called optimal coefficients (or according to E. Hlawka good lattice points) estimations of the type

$$R_N = O\left(\frac{\ln^{\alpha s} N}{N^\alpha}\right) \tag{16}$$

for functions $f \in E_s^\alpha(C)$, $\alpha > 1$. These estimations are better than the correspondent estimations for irrational k-Theta-sequences in 2.1.1. Unfortunately the good lattice points $a_1(N), \ldots, a_s(N)$ are in general not easily obtained.

2.1.3 Lattices according to A. Weil

They have the form

$$\underline{x}^k = \left(\left\{ \frac{ak}{N} \right\}, \left\{ \frac{ak^2}{N} \right\}, \ldots, \left\{ \frac{ak^s}{N} \right\} \right) \qquad (a, N) = 1,\ k = 1, \ldots, N \tag{17}$$

We call this lattice an A. Weil lattice, because A. Weil did some very important work on the estimation of the corresponding trigonometric sums. For the Weil-Lattice [3, 5] one can show for $f \in E_s^\alpha(C)$, $\alpha > 1$

$$R_N = O\left(\frac{1}{\sqrt{N}}\right) \tag{18}$$

whereas the term $1/\sqrt{N}$ being independent on α and the dimensionality s of the problem. Obviously the convergence rate $1/\sqrt{N}$ is rather poor and the estimation is not sharp even in the case of dimensionality being $s = 1$. On the other hand the Weil-Lattice has a somewhat more random behaviour than the previous lattices which represent (semi)group structures in I^s. Weil-Lattices are somewhat more attractive for statisticians.

2.1.4 The Orthogonal Lattice

$$\underline{x}^k = \underline{x}(k_1, \ldots, k_s) = \tfrac{1}{M}(k_1, \ldots, k_s) \qquad \text{each } k_j \text{ running from } 0 \text{ to } M-1$$
$$j = 1, \ldots, s,\ N = M^s \tag{19}$$

An unequivocal mapping from k to $(k_1, \ldots, k_s)$ can be given if k is represented in Base M.

The orthogonal lattices represent obviously the cartesian product of one dimensional rules. The number of nodes $N = M^s$ in orthogonal lattices increases very rapidly with increasing dimension s. On the other hand the accuracy of the s-dimensional cartesian product rule will not be better than in the one dimensional case. This heuristic observation is the main reason for the poor (but best possible) estimation for cartesian product rules on the classes $E_s^\alpha(C)$:

$$R_N = O\left(\frac{1}{N^{\alpha/s}}\right) \tag{20}$$

Almost only for historical reasons and for the sake of respectfulnes we considered and implemented this type of integration rules.

In forthcoming papers we will consider and implement further lattice rules and the (t,m,s)-nets as well. For the definition of those nets see [4].

2.2 Periodization

The integration rules described above heavily depend on the periodicity of the functions to be integrated numerically. Because the assumption of periodicity is not fulfilled in most practical cases the applicability of NTN seems to be very limited at the first glance. Fortunately there are methods which transform non periodic functions into periodic functions in an integral preserving way.

A first simple example of such a transform:
If $f(x), 0 \le x \le 1$, is continous then the function

$$g(x) = \frac{1}{2}(f(x) + f(1-x)) \tag{21}$$

can be extended to the real line in a continous and 1-periodic manner. Obviously

$$\int_0^1 g(x)\,dx = \int_0^1 f(x)\,dx \tag{22}$$

holds as well. This method can easily be modified for functions of several variables too. Unfortunately this method does not regard differentiability at the endpoints of the intervals. So methods of this type are most preferable in the case of nonsmooth integrands.

Following Korobov [3] we give a more general definition of periodization operators:

Let $\psi(x) : [0,1] \to [0,1]$ an increasing function with continous derivatives up to an order $\alpha + 1$. Let further $\psi(0) = 0, \psi(1) = 1, \psi^{(n)}(0) = \psi^{(n)}(1) = 0, n = 1, \ldots, \alpha + 1$. Then we call ψ a periodization function of order α.

A periodization of order α turns a function with partial derivatives of order α into a periodic function with the same differentiability properties and furthermore into a function belonging to $E_s^\alpha(C)$. This fact is treated in full details in [2, 3].

In this case we have

$$\int_{I^s} f(x_1, \ldots, x_s)\,dx_1 \cdots dx_s = \int_{I^s} f(\psi(t_1), \ldots, \psi(t_s))\psi'(t_1) \cdots \psi'(t_s)\,dt_1 \cdots dt_s \tag{23}$$

and the integrand and its derivatives up to an order α can be extended periodically to the whole space $\mathbb{R}^s$.

These mathematically simple transforms turn in theory and in practice slow convergent rules frequently into fast convergent ones. See for example [3].

We give two examples of periodization functions in the subsequent chapter three. In a subsequent paper new “universal” periodization operators and operators for integrands having singularities will be introduced. Obviously methods of this type don’t affect the parallelizability of the integration rules.

3 Implementation

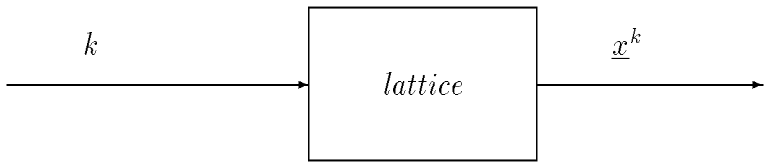
In this chapter we want to give a view on the program components, the implemented methods, the parallelization and some timing results.

The integral is approximated in the following way:

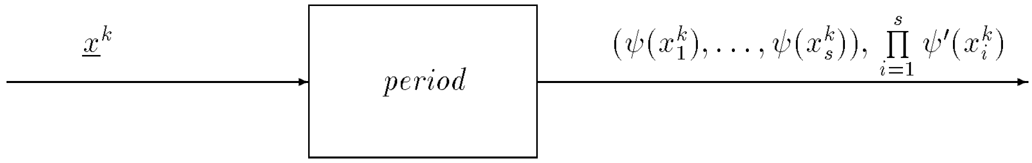
$$\int_{I^s} f(x_1, \ldots, x_s)\, dx_1 \cdots dx_s + R_N(f) = \frac{1}{\sum \rho_k} \sum_{k=1}^{N} (\rho_k\, f(\psi(x_1^k), \ldots, \psi(x_s^k)) \prod_{i=1}^{s} \psi'(x_i^k)) \tag{24}$$

For each term of the sum (24) the following five steps are executed:

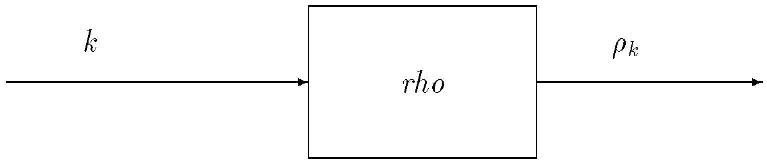
1. evaluation of the k-th lattice point:



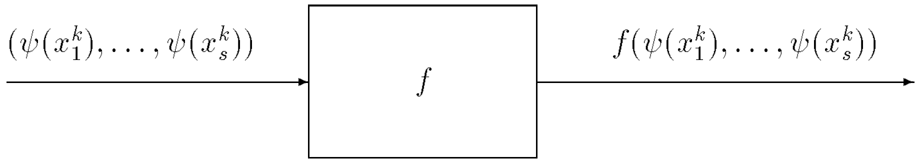
2. periodization:



3. computation of weights:



4. evaluation of function value:



5. evaluate
$$\rho_k\, f(\psi(x_1^k), \ldots, \psi(x_s^k)) \prod_{i=1}^{s} \psi'(x_i^k)$$

3.1 Subroutines

3.1.1 Subroutine lattice

The lattices that can be computed in the current implementation are the four described in chapter 2:

1. **k-theta-sequence:** $\underline{x}^k = (\{k\Theta_1\}, \{k\Theta_2\}, \ldots, \{k\Theta_s\})$
The vector $(\Theta_1, \ldots, \Theta_s)$ is set to the fixed value $(\exp(1), \ldots, \exp(s))$.
2. **parallelepipedic lattice:** $\underline{x}^k = (\left\{\frac{a_1 k}{N}\right\}, \left\{\frac{a_2 k}{N}\right\}, \ldots, \left\{\frac{a_s k}{N}\right\})$
The optimal coefficients $a_1, \ldots, a_s$ must be given as input to the program, because they depend both on the number of lattice points N and on the dimension s and are not computed here. Sets of optimal coefficients are tabulated for example in [2, 3]. In forthcoming papers of the RIST++ new parallel methods for the computation of optimal coefficients and more extensive tables which meet the contemporary demands of high performance scientific computation will be published.
3. **A. Weil-lattice:** $\underline{x}^k = (\left\{\frac{ak}{N}\right\}, \left\{\frac{ak^2}{N}\right\}, \ldots, \left\{\frac{ak^s}{N}\right\})$
The integer a with $(a, N) = 1$ has to be given as input.
4. **orthogonal lattice:** $\underline{x}^k = \frac{1}{M}(k_1, k_2, \ldots, k_s)$.
The k_j are computed from k by the property $k = \sum_{j=1}^{s} k_j M^{j-1}$, $M^s = N$
(i. e. the k_j are the digits of k represented in Base M.)

3.1.2 Subroutine period

If no periodization is wanted, this subroutine returns $\underline{x}^k$ unchanged and sets $\prod_{i=1}^{s} \psi'(x_i^k) = 1$.

The available periodization functions are:

1. **Algebraic Beta Periodization:**

$$\psi_{\alpha,\beta}(t) = B(\alpha, \beta, t) = \frac{\Gamma(\alpha+\beta+2)}{\Gamma(\alpha+1)\Gamma(\beta+1)} \int_0^t \tau^\alpha (1-\tau)^\beta \, d\tau \tag{25}$$

In this program only the case $\alpha = \beta \in \mathbb{N}$ for which $B(\alpha, \alpha, t)$ is a polynomial of degree $2\alpha + 1$ is implemented.

$$\psi_\alpha(t) = B(\alpha,\alpha,t) = (2\alpha+1)\binom{2\alpha}{\alpha}\sum_{j=0}^{\alpha}\binom{\alpha}{j}(-1)^j\frac{t^{\alpha+j+1}}{\alpha+j+1} \tag{26}$$

$$\psi'_\alpha(t) = (2\alpha+1)\binom{2\alpha}{\alpha}(t(1-t))^\alpha \tag{27}$$

2. **Trigonometric Periodization:**

$$\psi_\alpha(t) = c_\alpha \int_0^t \sin^\alpha \pi\tau \, d\tau \qquad c_\alpha = (\int_0^1 \sin^\alpha \pi\tau \, d\tau)^{-1} \tag{28}$$

Which can be written (and is computed) as:

$$\begin{aligned}\psi_\alpha(t) = c_\alpha\,(& -\frac{\sin^{\alpha-1}\pi t\cos\pi t}{\alpha\pi} - \frac{(\alpha-1)\sin^{\alpha-3}\pi t\cos\pi t}{\alpha(\alpha-2)\pi} - \\ & -\cdots- \\ & -\frac{(\alpha-1)(\alpha-3)\cdots5\cdot3\sin\pi t\cos\pi t}{\alpha(\alpha-2)\cdots4\cdot2\pi} + \\ & +\frac{(\alpha-1)(\alpha-3)\cdots5\cdot3t}{\alpha(\alpha-2)\cdots4\cdot2})\end{aligned} \tag{29}$$

$$c_\alpha = \frac{\alpha(\alpha-2)\cdots4\cdot2}{(\alpha-1)(\alpha-3)\cdots5\cdot3} \qquad \alpha = 2n, n\in\mathbb{N} \tag{30}$$

$$\begin{aligned}\psi_\alpha(t) = c_\alpha\,(& -\frac{\sin^{\alpha-1}\pi t\cos\pi t}{\alpha\pi} - \frac{(\alpha-1)\sin^{\alpha-3}\pi t\cos\pi t}{\alpha(\alpha-2)\pi} - \\ & -\cdots- \\ & -\frac{(\alpha-1)(\alpha-3)\cdots4\cdot2\cos\pi t}{\alpha(\alpha-2)\cdots5\cdot3\pi} +\end{aligned}$$

$$+\frac{(\alpha-1)(\alpha-3)\cdots 4\cdot 2}{\alpha(\alpha-2)\cdots 5\cdot 3\pi}) \tag{31}$$

$$c_\alpha = \frac{\pi}{2}\cdot\frac{\alpha(\alpha-2)\cdots 5\cdot 3}{(\alpha-1)(\alpha-3)\cdots 4\cdot 2} \qquad \alpha = 2n+1, n\in\mathbb{N} \tag{32}$$

$$\psi'_\alpha(t) = c_\alpha \sin^\alpha \pi t \tag{33}$$

For both periodizations the constants c_α and the coefficients for the sums are computed once in the setup routine.

3.1.3 Subroutine rho

If no weights are selected the subroutine returns 1 else it calls the subroutine bakhv where the Bakhvalov weights are computed.

Weights of four different orders t can be selected: $t \in \{2,3,4,5\}$

Definition and Properties of Bakhvalov Weights[1]

- As defined in chapter 2 the k-th Bakhvalov weight of order $t = \rho^t_{k,M}$ is the coefficient of x^{k-1} in $(1+x+\cdots+x^M)^t$. It can be seen that there are $Mt+1$ such coefficients. Therefore for a given number of lattice points N when t is selected, M is set to the smallest integer satisfying $Mt+1 \geq N$.
- Symmetry:
  $\rho^t_{k,M} = \rho^t_{Mt+2-k,M} \qquad k = 1,2,\ldots,Mt+1$
- Sum of weights:
  $\sum_{k=1}^{Mt+1} \rho^t_{k,M} = (M+1)^t$
- Recursive computation:
  $\rho^1_{k,M} = 1 \qquad k = 1,2,\ldots,M+1$
  $\rho^{t+1}_{k,M} = \sum_{i=\max(1,k-M)}^{\min(Mt+1,k)} \rho^t_{i,M} \qquad k = 1,2,\ldots,M(t+1)+1$

  (Using this formula in the program results in inacceptable computation time!)

In the program the following formulas which use the symmetry property are implemented:

$$\rho^2_{k,M} = k \qquad {\scriptstyle k=1,\ldots,M+1} \tag{34}$$

$$\rho^3_{k,M} = \binom{k+1}{2} \qquad {\scriptstyle k=1,\ldots,M+1} \tag{35}$$

$$= \binom{k+1}{2} - 3\binom{k-M}{2} \qquad {\scriptstyle k=M+2,\ldots,\lfloor \frac{3M+1}{2} \rfloor} \tag{36}$$

$$\rho^4_{k,M} = \frac{k(k+1)(k+2)}{6} \qquad {\scriptstyle k=1,\ldots,M+1} \tag{37}$$

$$= \frac{k(k+1)(k+2)}{6} - 4\frac{(k-M-1)(k-M)(k-M+1)}{6} \qquad {\scriptstyle k=M+2,\ldots,2M+1} \tag{38}$$

$$\rho^5_{k,M} = \frac{k(k+1)(k^2+5k+6)}{24} \qquad {\scriptstyle k=1,\ldots,M+1} \tag{39}$$

$$= \frac{k(k+1)(k^2+5k+6)}{24} - - \frac{5(k-M-1)(k-M)((k-M-1)^2}{24} + \frac{25(k-M-1)+6)}{24} \qquad {\scriptstyle k=M+2,\ldots,2M+2} \tag{40}$$

$$= \frac{6k^4 - 12k^3(5M+2) + 6k^2(35M^2+25M+1)}{24} - - \frac{6k(50M^3+45M^2-5M-6)}{24} + + \frac{5M(31M^3+34M^2-7M-10)}{24} \qquad {\scriptstyle k=2M+3,\ldots,\lfloor \frac{5M+2}{2} \rfloor} \tag{41}$$

In those cases where $Mt + 1 > N$ the sum of the used weights is $< (M+1)^t$. This is corrected in the main routine.

All the above subroutines can be easily extended to handle further integration rules.

3.1.4 Integrand f

The integrand function f has to be programmed as an own subroutine.

3.2 Parallelization with PVM

The structure of the problem permits a coarse grain parallelization. The basic idea is to split the computation of the sum (24) and to evaluate the parts in parallel. As each term of the sum is independent of all others, the sum can be split into arbitrary parts. On a homogenous network where the user disposes of 100% of the machine capacity on each node this could be done by partitioning of the sum into one part for each node. As it is shown later with some timing results this method achieves an almost linear speedup.

If there are other processes running on some nodes or a heterogenous network is used, a processor with a high workload or a slower one would cause a delay in the whole computation. With a finer partition of the problem it is possible to react on changes in the systems performance. Therefore the integration program is written in a master-slave style. There is one masterprocess (the program INTEGRATE) which controls the slaveprocesses (INTSLAVE) on each PVM node. When a slaveprocess has computed a partial result the value is returned to the masterprocess, which sends a new task to the slaveprocess.

The number of tasks must be specified by the user. Here two facts should be taken into account:

- A large number of tasks (i. e. small tasks) causes less idle times at the end of the computation (when only few of the processors have not yet finished their task).
- On the other hand the communication time rises with the number of tasks.

Some experiments on the DEC-ALPHA Cluster of the RIST++ showed, that the communication time increases linearily with the number of tasks - provided that the computation time of the task remains greater than 0.5 seconds.

Structure of the masterprogram:

enroll in pvm
get startparameters
get number of available processors n
get number of subproblems k
invoke n slaves
send parameters
send first n subproblems to slaves
loop i = 1 to k
    receive partial result from a slave
    if $(i \leq k - n)$ *send next part to that slave*
end loop
print output
terminate slaves
exit

Structure of the slaveprogram:

enroll in pvm
receive parameters
'infinite'loop
    receive part of work
    compute partial result
    send result to master
end loop

When there is no further work, the slaveprocess blocks at the receive-statement until it is terminated by the master.

3.3 Timing results

The following measurements were made on ten DEC 5000/33 Workstations. The integrand function was

$$f(x_1, \ldots, x_s) = [\prod_{i=1}^{s} 1 + 2\pi^2(x_i^2 - x_i + \frac{1}{6})] - 1 \tag{42}$$

with

$$\int_{I^s} f(x_1, \ldots, x_s)\, dx_1 \cdots dx_s = 0 \qquad \forall s \in \mathbb{N} \tag{43}$$

Example 1
Integration parameters:

- dimension: 5
- lattice: parallelepipedic
- number of points: 374181
- opt. coefficients: 1, 343867, 255381, 310881, 115892
- periodization: algebraic, $\alpha = 6$
- weights: no
- number of partitions: equal to number of processors

Result of the integration: 3.897855 E-8

# nodes	1	2	3	4	5	6	7	8	9	10
time*(sec)	92.6	46.6	31.1	23.7	19.1	16.2	14.0	12.4	11.2	10.3
speedup*		1.99	2.96	3.91	4.85	5.72	6.61	7.47	8.27	8.99

Example 2
Integration parameters:

- dimension: 5
- lattice: k-theta-sequence
- number of points: 1500000
- periodization: algebraic, $\alpha = 2$
- weights: Bakhvalov-weights, $t = 5$
- number of partitions: equal to number of processors

Result of the integration: -6.9038953 E-8

# nodes	1	2	3	4	5	6	7	8	9	10
time*(sec)	225.6	113.6	76.3	57.1	45.6	37.9	32.9	28.9	25.9	23.5
speedup*		1.99	2.96	3.95	4.95	5.95	6.88	7.81	8.74	9.60

*Time is the wallclocktime of the master process. The time for one node is also the time of the parallel code, speedup values refer to this time.

As the table shows and as from the structure of the problem could be expected the speedup is quite linear for “large” problem sizes.

4 Conclusion and Future Work

The concept of PVM (which is a de-facto standard in parallel computation) meets especially on workstation-farms the demands for cheap computation power for high performance scientific compututation. So this PVM-oriented implementation of these – from the mathematical point of view best possible – numerical methods widens significantly the range of practical computability of high dimensional integrals.

In forthcoming work RIST++ will develop and implement a wide range of number theoretical numerics. This collection of algorithms (NEWTON) will be available via FTP.

References

[1] N. S. Bakhvalov *Optimal convergence bounds for quadrature processes and integration methods of Monte Carlo type for classes of functions,* Z. Vycisl. Mat. i Mat. Fiz., 4, suppl., (1964), pp. 5-63

[2] Hua Loo Keng and Wang Yuan *Applications of Number Theory to Numerical Analysis*, Springer Verlag Berlin Heidelberg New York, Science Press, Beijing (1981).

[3] N. M. Korobov, *Number-Theoretic Methods in Approximate Analysis,* Fizmatgiz, Moscow, 1963. (In Russian)

[4] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods,* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.

[5] A. Weil, *On some exponential sums,* Proc. Nat. Acad. Sci. U.S.A. 34 (1948).

[6] P. Zinterhof, *Einige zahlentheoretische Methoden zur numerischen Quadratur und Interpolation,* Sitzungsber. Österr. Akad. Wiss. Math.-naturw. Kl. II, 177 (1969), pp. 51-77.

[7] ——, *Über einige Abschätzungen bei der Approximation von Funktionen mit Gleichverteilungsmethoden,* Sitzungsber. Österr. Akad. Wiss. Math.-Naturw. Kl. II, 185 (1976), pp. 121-132.

[8] ——, *Gratis lattice points for multidimensional integration,* Computing, 38 (1987), pp. 347-353.

[9] P. Zinterhof and H. Stegbuchner, *Trigonometrische Approximation mit Gleichverteilungsmethoden,* Studia Sci. Math. Hungar., 13 (1978), pp. 273-289.

[10] P. Zinterhof and M. Salchegger, *Numerical Inversion of Multivariate Laplace Transforms Using a Parallel System,* ACPC/TR 92-20, 1992.