IC-MSQUARE 2012: International Conference on Mathematical Modelling in Physical Sciences IOP Publishing
Journal of Physics: Conference Series **410** (2013) 012047 doi:10.1088/1742-6596/410/1/012047

Adaptive control in multi-threaded iterated integration

Elise de Doncker[1] **and Fukuko Yuasa**[2]

[1]Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008, U.S.A.

[2]High Energy Accelerator Research Organization (KEK), 1-1 Oho, Tsukuba, Ibaraki 305-0801, Japan

E-mail: elise.dedoncker@wmich.edu, fukuko.yuasa@kek.jp

Abstract. In recent years we have developed a technique for the direct computation of Feynman loop-integrals, which are notorious for the occurrence of integrand singularities. Especially for handling singularities in the interior of the domain, we approximate the iterated integral using an adaptive algorithm in the coordinate directions. We present a novel multi-core parallelization scheme for adaptive multivariate integration, by assigning threads to the rule evaluations in the outer dimensions of the iterated integral. The method ensures a large parallel granularity as each function evaluation by itself comprises an integral over the lower dimensions, while the application of the threads is governed by the adaptive control in the outer level. We give computational results for a test set of 3- to 6-dimensional integrals, where several problems exhibit a loop integral behavior.

1. Introduction and background

Automatic integration can be viewed as a black-box procedure for the computation of an integral

$$\mathcal{I}f = \int_{\mathcal{D}} f(\vec{x})\ d\vec{x},$$

which takes a user-specification of the problem as input, including the integrand function $f(\vec{x})$, the domain $\mathcal{D}$, the dimension N and a requested accuracy. The latter may be determined by a relative and/or absolute error tolerance, t_r and t_a, respectively, in order to impose a criterion of the form $|\,Qf - \mathcal{I}f\,| \;\leq\; Ef \;\leq\; \max\{\,t_a\,,\,t_r\,|\mathcal{I}f\,|\,\}$, where Qf represents the obtained integral approximation and Ef is an estimate of the actual absolute error $|Qf - \mathcal{I}f|$.

Let us consider an *adaptive* integration procedure that adheres to the meta-algorithm of Figure 1. By domain partitioning it is intended to sample the integrand intensively in the vicinity of problematic function behavior. This type of algorithm maintains a priority queue data structure (such as a heap or a linked list) on the set of subregions, which is generally keyed with the estimated error over the subregions. At each step, the highest priority region is selected, deleted from the priority queue and subdivided. The local integration is performed over the new subregions, which in turn are added to the priority queue.

The 1D general adaptive QUADPACK algorithms [1] use Kronrod rules for the local integration, and generate local error estimates based on the Kronrod approximation and its corresponding

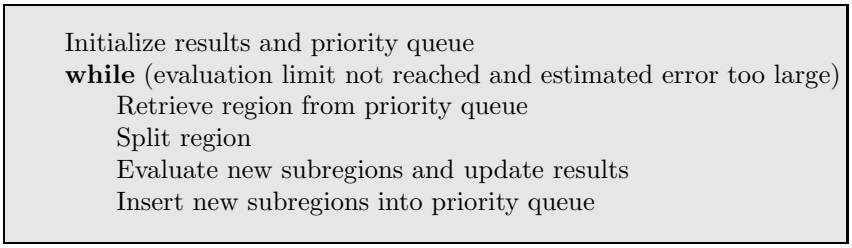


Figure 1. Adaptive integration meta-algorithm

Guass rule sum. As another example, the multivariate adaptive routine DCUHRE [2] calculates a nested family of rules for the integration and error estimation on each subregion.

We calculate an iterated integral, as formulated in Section 2, where the numerical integration on each level of the iterated integral is performed adaptively and according to the algorithm description of Figure 1. A parallel algorithm is obtained by evaluating the integration rule points in parallel in the outer level. In Section 3 we give results and discuss the parallel performance of the method for a set of sample problems, including the "DICE" integrals which characterize the singular behavior of some classes of Feynman loop integrals. The latter occur in perturbation calculations for the cross section of particle interactions in high-energy physics. This paper presents an extension of the work described in [3] (which was restricted to three dimensions).

2. Parallel iterated integration

Consider an *iterated (repeated)* integral

$$\mathcal{I}f = \int_{\mathcal{D}_1} d\vec{x}^{(1)} \ldots \int_{\mathcal{D}_\ell} d\vec{x}^{(\ell)}\ f(\vec{x}^{(1)}, \ldots, \vec{x}^{(\ell)})$$

over the N-dimensional product region $\mathcal{D} = \mathcal{D}_1 \times \ldots \times \mathcal{D}_\ell$, where the $\mathcal{D}_j,\ \ j = 1, \ldots, \ell$ may be 1D or multidimensional regions with associated (for example, adaptive) integration methods.

For an iterated integration over $\mathcal{D} = \mathcal{D}_1 \times \mathcal{D}_2$, the integral over a selected subregion $S \subseteq \mathcal{D}_1$ is approximated numerically by a (scaled) integration rule with points $\vec{x}_k$ and weights w_k, $k = 1, \ldots, K$, of the form

$$\int_S d\vec{x}^{(1)}\ F(\vec{x}^{(1)}) \ \approx\ \sum_{k=1}^{K} w_k\, F(\vec{x_k}^{(1)}), \quad \text{where}\ \ F(\vec{x_k}^{(1)}) \ =\ \int_{\mathcal{D}_2} d\vec{x}^{(2)}\ f(\vec{x_k}^{(1)}, \vec{x}^{(2)}).$$

The values of F are themselves integrals and they are independent from each other; hence they can be evaluated in parallel. The weighted sum referred to as the (outer) integration rule sum is thus calculated by parallel threads, over each subregion $S \subseteq \mathcal{D}_1$ generated by the adaptive partitioning strategy in the outer level of the iterated integral.

Our current parallel implementation uses successive one-dimensional quadratures, over the given regions (intervals) $\mathcal{D}_j$, for $j = 1, \ldots, \ell = N$. For the integration in each coordinate direction we use a general adaptive program (DQAGE) from the QUADPACK package [1]. The interval in each coordinate direction is partitioned adaptively according to the algorithm of Figure 1. For the local integral and error estimates over the subintervals, a pair of quadrature formulas is applied consisting of a Gauss rule and its corresponding Kronrod formula.

The test results in the next section are obtained by evaluating the outer integral in parallel using the 21-point Gauss-Kronrod pair over the subintervals generated by the adaptive strategy. For the inner integrations, the 21-point or the 15-point pair can be selected by the user.

3. Parallel integration results

A set of simple test integrals of dimensions 1 through 6 is given in formulas (1)-(10) below. Integrals (1)-(3) are 3-dimensional, (4)-(6) are 4-dimensional, (7) and (8) are 5-dimensional, (9) and (10) are 6-dimensional sample problems. Various problem characteristics are represented, including (very) high-degree polynomials in (4) and (10), and an oscillating function in (8). The integrands of (3) and (6) have a corner singularity which requires intensive local partitioning around a single point of the domain.

$$\mathcal{I}f_1 = \int_{-1}^{1} dx_1 \int_{-1}^{1} dx_2 \ldots \int_{-1}^{1} dx_N \; \frac{\delta\, r\; \theta(1-r^2)}{(r^2-\alpha^2)^2+\delta^2}, \tag{1}$$

$$\text{where} \;\; r = \Big(\sum_{j=1}^{N} x_j^2\Big)^{1/2}, \;\; N=3, \;\; \delta = 10^{-6}, \;\; \alpha = 0.8$$

$$\text{and} \;\; \theta(\chi) = 1 \;\; (\text{if } \chi > 0); \;\; 1/2 \;\; (\text{if } \chi = 0); \;\; 0 \;\; (\text{if } \chi < 0)$$

$$\mathcal{I}f_2 = \int_{-1}^{1} dx_1 \int_{-1}^{1} dx_2 \ldots \int_{-1}^{1} dx_N \; \frac{2\,\delta\, x_2}{(\sum_{j=1}^{N} x_j - 1)^2+\delta^2}, \quad \text{with } N=3, \;\; \delta = 10^{-6} \tag{2}$$

$$\mathcal{I}f_3 = \int_{0}^{1} dx \int_{0}^{1} dy \int_{0}^{1} dz \; \frac{1}{(x+y+z)^2} \tag{3}$$

$$\mathcal{I}f_4 = \int_{0}^{1} dx \int_{0}^{1} dy \int_{0}^{1} dz \int_{0}^{1} du \; (0.66\times 10^7)\; x^{54}\, y^{39}\, z^{59}\, u^{49} \tag{4}$$

$$\mathcal{I}f_5 : \;\; \text{as} \;\; \mathcal{I}f_2 \;\; \text{for } N=4, \;\; \text{with } \delta = 10^{-6} \tag{5}$$

$$\mathcal{I}f_6 = \int_{0}^{1} dx \int_{0}^{1} dy \int_{0}^{1} dz \int_{0}^{1} du \; \frac{1}{(x+y+z+u)^2} \tag{6}$$

$$\mathcal{I}f_7 : \;\; \text{as} \;\; \mathcal{I}f_1 \;\; \text{for } N=5, \;\; \text{with } \delta = 10^{-1} \tag{7}$$

$$\mathcal{I}f_8 = \int_{0}^{\pi} dx \int_{0}^{\pi} dy \int_{0}^{\pi} dz \int_{0}^{\pi} du \int_{0}^{\pi/2} dv \;\; \cos(6(x+y+z+u+v)) \tag{8}$$

$$\mathcal{I}f_9 : \;\; \text{as} \;\; \mathcal{I}f_1 \;\; \text{for } N=6, \;\; \text{with } \delta = 10^{-1} \tag{9}$$

$$\mathcal{I}f_{10} = \int_{0}^{1} dx \int_{0}^{1} dy \int_{0}^{1} dz \int_{0}^{1} du \int_{0}^{1} dv \int_{0}^{1} dw \; (3.78\times 10^9)\; x^{49}\, y^{39}\, z^{44}\, u^{29}\, v^{39}\, w^{34} \tag{10}$$

The integral (1) was used in [4] with $N = 3$, as a test problem (named "DICE3") for the Monte Carlo integration program DICE, and (2) is a 3D version of the problem "DICE1", treated for $N = 2$ in [4]. A two-dimensional version of the integrand f_1 (as a function of x_1 and x_2 over $[-1, 1]^2$) is depicted in the plot of Figure 2 for $\delta = 0.1$ and $\alpha = 0.8$ [5]. The function has a discontinuity on the unit circle (outside of which it falls to zero), and a ridge at $r^2 = \alpha^2 = 0.64$, with a peak height of α/δ. Thus the ridge inside the circle is steep and narrow when delta is small (e.g., $\delta = 10^{-6}$ in (1)). Both the discontinuity and the ridge present numerical integration difficulties, occurring in the interior of the domain. The integrand of problem (2) has a peaked behavior at $\sum_{j=1}^{N} x_j = 1$, which is more pronounced for smaller δ (note $\delta = 10^{-6}$ in (2)). The peaked behavior of the integrand functions in (1) and (2) mimics the irregular behavior of the integrands of some classes of Feynman loop integrals in high energy physics.

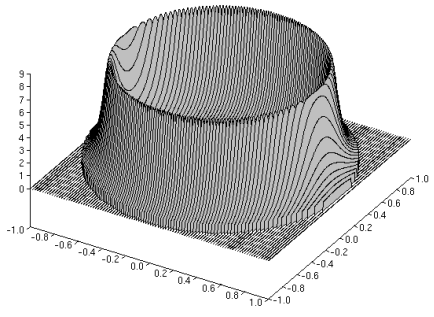
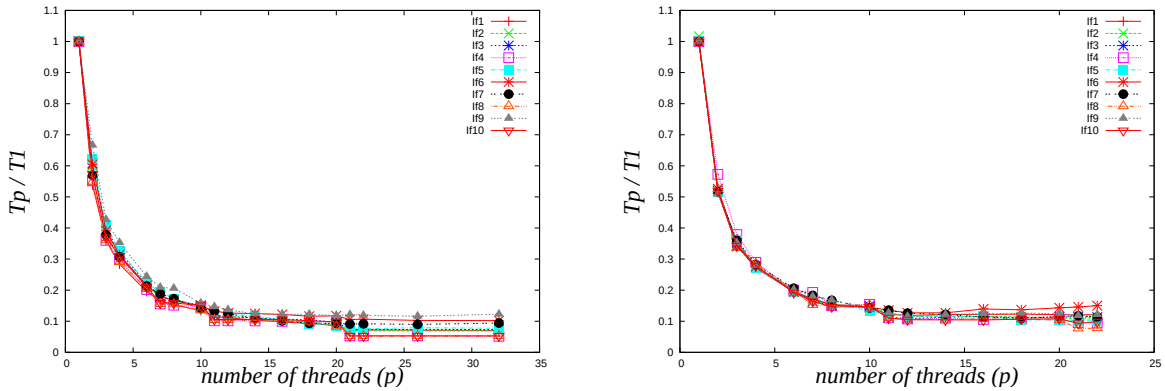


Figure 2. Plot of integrand $f_1(\vec{x})$ of (1) for $N = 2, \;\; \delta = 0.1, \;\; \alpha = 0.8$



(a) (b)

Figure 3. ρ_p vs. number of threads p : (a) on SR16000; (b) on Intel Xeon workstation

Particularly in relatively low dimensions ($N = 2, 3, 4$, say), the ability of the iterated adaptive integration method to achieve high accuracy for very small δ is often remarkable, compared to, e.g., slowly converging Monte-Carlo (MC) methods and standard multivariate adaptive subdivision strategies. In [5], results were given for DICE3 ($N = 3$, $\delta = 10^{-m}$, $1 \le m \le 6$). For example, a simple MC method with antithetic variates (RCRUDE [6]), given a maximum of $100\,M = 10^8$ points, yields results to relative accuracies of 0.013 and 0.017 for $\delta = 10^{-5}$ and $\delta = 10^{-6}$, respectively. For the same δ values, the standard global adaptive (GA) algorithm of DCUHRE [2] using $250\,M$ points has no correct digits in the results, with relative errors of 0.65 and 0.95. However, the iterated integration method can achieve high requested accuracies (e.g., 10 correct digits) for these parameter values, at the expense of function evaluations but with only a fraction of the memory usage of standard GA.

Parallel performance results of our multi-threaded (OpenMP) implementation of the code for the test problems (1)-(10) are shown in Figures 3(a) and 3(b). Each curve represents the ratio $\rho_p = T_p/T_1$ as a function of the number of threads p. Here T_1 is the sequential time and T_p is the execution time when using p threads; thus ρ_p is the parallel time scaled by T_1.

Figure 3(a) is obtained from timing runs on a node of the Hitachi SR16000 system at KEK (with a Power7@3.83GHz CPU), and using the f90 compiler. Correspondingly, Figure 3(b) shows results obtained on a 12-core workstation (with dual 6-core Intel Xeon E5645 processors), and using gfortran for the compilation. Generally, good speedups are observed for our functions on both systems. The difficulty of the corner point singularity is apparent in the timing results for (4D) problem (6) on the 12-core Intel machine. Timings on this system show the 12-core limit; however, speedup is achieved for some functions to higher numbers of threads through hyper-threading. On the Hitachi SR16000 node, the current version of our code shows the effect of the parallelization based on the sample 21-point rule in the outer layer of the iterated integral, even though some of the times still decrease slightly for more threads.

The parallelization can easily be extended to the simultaneous integration over the two (or more) subregions obtained in the subdivision (*Split*) step of the meta-algorithm in Figure 1. We will further investigate an extension of the scheme allowing for massive parallelism by implementing parallelization in more than one, e.g., the outer two levels of the iterated integral.

4. Conclusions

We presented a novel parallel algorithm for iterated adaptive multivariate integration. Iterated multivariate integration in relatively low dimensions is superior to MC and GA with respect to accuracy, for problems exhibiting the behavior of some classes of Feynman loop integrals. Good

speedups of the parallel scheme are achieved on the SR16000 Hitachi system at KEK, Japan, and on a 12-core Intel (Xeon processor) based workstation. In future work we plan to target the parallelization to systems with many cores.

Acknowledgments
We acknowledge the support from the National Science Foundation under MRI Award Number 1126438, and from Western Michigan University and the College of Engineering and Applied Sciences (CEAS). In Japan, this work is supported by Grant-in-Aid for Scientific Research (24540292) of JSPS and the Large Scale Simulation Program No.12-07 of KEK.

References

[1] Piessens R, de Doncker E, Überhuber C W and Kahaner D K 1983 *QUADPACK, A Subroutine Package for Automatic Integration* Springer Series in Computational Mathematics (Springer-Verlag)
[2] Berntsen J, Espelid T O and Genz A 1991 *ACM Transactions on Mathematical Software* 17 452–456 URL `http://www.sci.wsu.edu/math/faculty/genz/homepage`
[3] de Doncker E and Yuasa F 2012 *Proc. of the Conference on Computational Physics, Oak Ridge* (The Journal of Physics: Conference Series) to appear
[4] Tobimatsu K and Kawabata S 1998 Multi-dimensional integration routine DICE Tech. Rep. 85 Kogakuin University
[5] de Doncker E, Kaugars K, Cucos L and Zanny R 2001 *Proc. of Computational Particle Physics Symposium (CPP 2001)* pp 110–119
[6] Genz A 1998 MVNDST URL `http://www.sci.wsu.edu/math/faculty/genz/homepage`