Vegas Revisited: Adaptive Monte Carlo Integration Beyond Factorization

Thorsten Ohl[*][†]

Darmstadt University of Technology
Schloßgartenstr. 9
D-64289 Darmstadt
Germany

IKDA 98/15
hep-ph/9806432
June 1998

Abstract

We present a new adaptive Monte Carlo integration algorithm for ill-behaved integrands with non-factorizable singularities. The algorithm combines Vegas with multi channel sampling and performs significantly better than Vegas for a large class of integrals appearing in physics.

1 Introduction

Throughout physics, it is frequently necessary to evaluate the integral $I(f)$ of a function f on a manifold M using a measure μ

$$I(f) = \int_M \mathrm{d}\mu(p)\, f(p)\,. \tag{1}$$

---

[*] e-mail: `ohl@hep.tu-darmstadt.de`

[†] Supported by Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie, Germany.

More often than not, an analytical evaluation in terms of elementary or known special functions is impossible and we have to rely on numerical methods for estimating $I(f)$. A typical example is given by the integration of differential cross sections on a part of phase space to obtain predictions for event rates in scattering experiments.

In more than three dimensions, standard quadrature formulae are not practical and Monte Carlo integration is the only option. As is well known, $I(f)$ is estimated by

$$E(f) = \left\langle \frac{f}{g} \right\rangle_g = \frac{1}{N} \sum_{i=1}^{N} \frac{f(p_i)}{g(p_i)} \,, \tag{2}$$

where g is the probability density (with respect to the measure μ) of the randomly distributed p_i, e. g. $g(p) = 1/\operatorname{Vol}(M)$ for uniformly distributed p_i. The error of this estimate is given by the square root of the variance

$$V(f) = \frac{1}{N-1} \left(\left\langle \left(\frac{f}{g} \right)^2 \right\rangle_g - \left\langle \frac{f}{g} \right\rangle_g^2 \right) \tag{3}$$

which suggests to choose a g that minimizes $V(f)$. If f is a wildly fluctuating function, this optimization of g is indispensable for obtaining a useful accuracy. Typical causes for large fluctuations are integrable singularities of f or μ inside of M or non-integrable singularities very close to M. Therefore, we will use the term “singularity” for those parts of M in which there are large fluctuations in f or μ.

Manual optimization of g is often too time consuming, in particular if the dependence of the integral on external parameters (in the integrand and in the boundaries) is to be studied. Adaptive numerical approaches are more attractive in these cases. The problem of optimizing g numerically has been solved for *factorizable* distributions g and measures μ by the classic Vegas [1] algorithm long ago. Factorizable g and μ are special, because the computational costs for optimization rise only linearly with the number of dimensions. In all other cases, there is a prohibitive exponential rise of the computational costs with the number of dimensions.

The property of factorization depends on the coordinate system, of course. Consider, for example, the functions

$$f_1(x_1, x_2) = \frac{1}{(x_1 - a_1)^2 + b_1^2} \tag{4a}$$

$$f_2(x_1, x_2) = \frac{1}{\left(\sqrt{x_1^2 + x_2^2} - a_2 \right)^2 + b_2^2} \tag{4b}$$

on $M = (-1,1) \otimes (-1,1)$ with the measure $\mathrm{d}\mu = \mathrm{d}x_1 \wedge \mathrm{d}x_2$. Obviously, f_1 is factorizable in Cartesian coordinates, while f_2 is factorizable in polar coordinates. Vegas will sample either function efficiently for arbitrary $b_{1,2}$ in suitable coordinate systems, but there is no coordinate system in which Vegas can sample the sum $f_1 + f_2$ efficiently for small $b_{1,2}$.

In this note, we present a generalization of the Vegas algorithm from factorizable distributions to sums of factorizable distributions, where each term may be factorizable in a *different* coordinate system. This larger class includes most of the integrands appearing in particle physics and empirical studies have shown a dramatic increase of accuracy for typical integrals. Technically, this generalization is the combination of the Vegas algorithm with adaptive multi channel sampling [2].

In section 2, we will discuss the coordinate transformations employed by the algorithm and in section 3, we will describe the adaptive multi channel algorithm. Finally, I will discuss the performance of a first implementation of the algorithm in section 4 and conclude.

2 Maps

The problem of estimating $I(f)$ can be divided naturally into two parts: parametrization of M and sampling of the function f. While the estimate will not depend on the parametrization, the error will.

In general, we need an atlas with more that one chart ϕ to cover the manifold M. We can ignore this technical complication in the following, because, for the purpose of integration, we can decompose M such that each piece is covered by a single chart. Moreover, a single chart suffices in most cases of practical interest, since we are at liberty to remove sets of measure zero from M. For example, after removing a single point, the unit sphere can be covered by a single chart.

Nevertheless, even if we are not concerned with the global properties of M that require the use of more than one chart, the language of differential geometry will allow us to use our geometrical intuition. Instead of pasting together locally flat pieces, we will paste together *factorizable* pieces, which can be overlapping, because integration is an additive operation.

For actual computations, it is convenient to use the same domain for the charts of all manifolds. The obvious choice for n-dimensional manifolds is the open n-dimensional unit hypercube

$$U = (0,1)^{\otimes n} \,. \tag{5}$$

Sometimes, it will be instructive to view the chart as a composition $\phi = \psi \circ \chi$

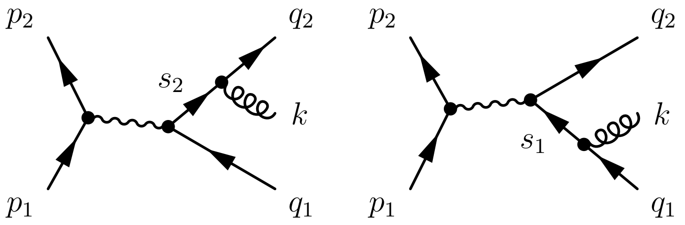

Figure 1: $e^+e^- \to q\bar{q}g$

with an irregularly shaped $P \in \mathbf{R}^n$ as an intermediate step

$$\begin{tikzcd}
& U \arrow[dl, "\chi"'] \arrow[d, "\phi"] \arrow[ddr, dotted, "f\circ\phi"] & \\
& M \arrow[dr, dotted, "f"] & \\
P \arrow[ur, "\psi"] \arrow[rr, dotted, "f\circ\psi"'] & & \mathbf{R}
\end{tikzcd} \tag{6}$$

(in all commutative diagrams, solid arrows are reserved for bijections and dotted arrows are used for other morphisms). The integral (1) can now be written

$$I(f) = \int_0^1 \mathrm{d}^n x \left|\frac{\partial\phi}{\partial x}\right| f(\phi(x)) \tag{7}$$

and it remains to sample $|\partial\phi/\partial x| \cdot (f \circ \phi)$ on U. Below, it will be crucial that there is more than one way to map U onto M

$$\begin{tikzcd}
U \arrow[rr, "\chi"] \arrow[dr, "\phi"] \arrow[dd, "\pi_U"'] & & P \arrow[dl, "\psi"'] \arrow[dd, "\pi_P"] \\
& M & \\
U' \arrow[ur, "\phi'"] \arrow[rr, "\chi'"] & & P' \arrow[ul, "\psi'"']
\end{tikzcd} \tag{8}$$

and that we are free to select the map most suitable for our purposes.

The ideal choice for ϕ would be a solution of the partial differential equation $|\partial\phi/\partial x| = 1/(f \circ \phi)$, but this is equivalent to an analytical evaluation of $I(f)$ and is impossible for the cases under consideration. A more realistic goal is to find a ϕ such that $|\partial\phi/\partial x| \cdot (f \circ \phi)$ has factorizable singularities and is therefore sampled well by Vegas. This is still a non-trivial problem, however.

For example, consider the phase space integration for gluon radiation $e^+e^- \to q\bar{q}g$. From the Feynman diagrams in figure 1 it is obvious that the squared matrix element will have singularities in the variables $s_{1/2} = (q_{1/2} +$

$k)^2$. Thus, adaptive sampling using Vegas would benefit from a parametrization using both s_1 and s_2 as coordinates in the intermediate space P. Unfortunately, the invariant phase space measure for such a parametrization involves the Gram determinant in the form $1/\sqrt{\Delta_4(p_1,p_2,q_1,q_2)}$, which will lead to non-factorizable singularities at the edges of phase space. Note that the very elegant phase space parametrizations of the RAMBO [3] type are not useful in this case, because there is no simple relation between the coordinates on U and the invariants in which the squared matrix elements can have singularities. On the other hand, it is straightforward to find parametrizations that factorize the dependency on s_1 or s_2 *separately*.

Returning to the general case, consider N_c different maps $\phi_i : U \to M$ and probability densities $g_i : U \to [0,\infty)$. Then the function

$$g = \sum_{i=1}^{N_c} \alpha_i (g_i \circ \phi_i^{-1}) \left| \frac{\partial \phi_i^{-1}}{\partial p} \right| \tag{9}$$

is a probability density $g : M \to [0,\infty)$

$$\int_M \mathrm{d}\mu(p)\, g(p) = 1\,, \tag{10}$$

as long as the g_i and α_i are properly normalized

$$\int_0^1 g_i(x) \mathrm{d}^n x = 1\,, \quad \sum_{i=1}^{N_c} \alpha_i = 1\,, \quad 0 \le \alpha_i \le 1\,. \tag{11}$$

From the definition (9), we have obviously

$$I(f) = \sum_{i=1}^{N_c} \alpha_i \int_M g_i(\phi_i^{-1}(p)) \left| \frac{\partial \phi_i^{-1}}{\partial p} \right| \mathrm{d}\mu(p)\, \frac{f(p)}{g(p)} \tag{12}$$

and, after pulling back from M to U

$$I(f) = \sum_{i=1}^{N_c} \alpha_i \int_0^1 g_i(x) \mathrm{d}^n x\, \frac{f(\phi_i(x))}{g(\phi_i(x))}\,, \tag{13}$$

we find the estimate

$$E(f) = \sum_{i=1}^{N_c} \alpha_i \left\langle \frac{f \circ \phi_i}{g \circ \phi_i} \right\rangle_{g_i}\,. \tag{14}$$

The factorized g_i in (12) and (14) can be optimized using the classic Vegas algorithm [1] unchanged. However, since we have to sample with a separate adaptive grid for each channel, a new implementation [4] is required for technical reasons.

Using the maps $\pi_{ij} = \phi_j^{-1} \circ \phi_i : U \to U$ introduced in (8), we can write the $g \circ \phi_i : U \to [0, \infty)$ from (14) as

$$g \circ \phi_i = \left| \frac{\partial \phi_i}{\partial x} \right|^{-1} \left(\alpha_i g_i + \sum_{\substack{j=1 \\ j \neq i}}^{N_c} \alpha_j (g_j \circ \pi_{ij}) \left| \frac{\partial \pi_{ij}}{\partial x} \right| \right) . \tag{15}$$

From a geometrical perspective, the maps π_{ij} are just the coordinate transformations from the coordinate systems in which the other singularities factorize into the coordinate system in which the current singularity factorizes.

Note that the integral in (12) does not change, when we use $\phi_i : U \to M_i \supseteq M$, if we extent f from M to M_i by the definition $f(M_i \setminus M) = 0$. This is useful, for instance, when we want to cover $(-1,1) \otimes (-1,1)$ by both Cartesian and polar coordinates. This causes, however, a problem with the π_{12} in (15). In the diagram

$$\begin{tikzcd}
P_1 \arrow[r, "\psi_1"] & M_1 & M \arrow[l, dotted, "\iota_1"'] \arrow[r, dotted, "\iota_2"] & M_2 & P_2 \arrow[l, "\psi_2"'] \\
& U \arrow[ul, "\chi_1"] \arrow[u, "\phi_1"'] \arrow[rr, dotted, "\pi_{12}"] & & U \arrow[u, "\phi_2"] \arrow[ur, "\chi_2"'] &
\end{tikzcd} \tag{16}$$

the injections $\iota_{1,2}$ are not onto and since π_{12} is not necessarily a bijection anymore, the Jacobian $|\partial \pi_{ij} / \partial x|$ may be ill-defined. But since $f(M_i \setminus M) = 0$, we only need the unique bijections $\phi'_{1,2}$ and π'_{12} that make the diagram

$$\begin{tikzcd}
P_1 \arrow[r, "\psi_1"] & M_1 & M \arrow[l, dotted, "\iota_1"'] \arrow[r, equal] & M \arrow[r, dotted, "\iota_2"] & M_2 & P_2 \arrow[l, "\psi_2"'] \\
& U \arrow[ul, "\chi_1"] \arrow[u, "\phi_1"'] & U_1 \arrow[l, dotted, "\iota_1^U"] \arrow[u, "\phi'_1"] \arrow[r, "\pi'_{12}"] & U_2 \arrow[u, "\phi'_2"'] \arrow[r, dotted, "\iota_2^U"'] & U \arrow[u, "\phi_2"] \arrow[ur, "\chi_2"'] &
\end{tikzcd} \tag{17}$$

commute.

In many applications, the dependence of an integral on external parameters has to be studied. Often, the π_{ij} will not depend on these parameters and we can rely on Vegas to optimize the g_i for each parameter set. In the next section, we will show how to optimize the α_i numerically as well.

3 Multichannel

Up to now, we have not specified the α_i, they are only subject to the conditions (11). Intuitively, we expect the best results when the α_i are proportional to the contribution of their corresponding singularities to the integral. The option of tuning the α_i manually is not attractive if the optimal values depend on varying external parameters. Instead, we use a numerical procedure [2] for tuning the α_i.

We want to minimize the variance (3) with respect to the α_i. This is equivalent to minimizing

$$W(f,\alpha) = \int_M g(p)\mathrm{d}\mu(p) \left(\frac{f(p)}{g(p)}\right)^2 \tag{18}$$

with respect to α with the subsidiary condition $\sum_i \alpha_i = 1$. After adding a Lagrange multiplier, the stationary points of the variation are given by the solutions to the equations

$$\forall i : W_i(f,\alpha) = W(f,\alpha) \tag{19}$$

where

$$W_i(f,\alpha) = -\frac{\partial}{\partial \alpha_i} W(f,\alpha) = \int_0^1 g_i(x)\mathrm{d}^n x \left(\frac{f(\phi_i(x))}{g(\phi_i(x))}\right)^2 \tag{20}$$

and

$$W(f,\alpha) = \sum_{i=1}^{N_c} \alpha_i W_i(f,\alpha)\,. \tag{21}$$

It can easily be shown [2] that the stationary points (19) correspond to local minima. If we use

$$N_i = \alpha_i N \tag{22}$$

to distribute N sampling points among the channels, the $W_i(f,\alpha)$ are just the contributions from channel i to the total variance. Thus we recover the familiar result from stratified sampling, that the overall variance is minimized by spreading the variance evenly among channels.

The $W_i(f,\alpha)$ can be estimated with very little extra effort while sampling $I(f)$ (cf. 14)

$$V_i(f,\alpha) = \left\langle \left(\frac{f\circ\phi_i}{g\circ\phi_i}\right)^2 \right\rangle_{g_i}\,. \tag{23}$$

Note that the factor of g_i/g from the corresponding formula in [2] is absent from (23), because we are already sampling with the weight g_i in each channel separately.

The equations (19) are a fixed point of the prescription

$$\alpha_i \mapsto \alpha_i' = \frac{\alpha_i \left(V_i(f,\alpha)\right)^\beta}{\sum_i \alpha_i \left(V_i(f,\alpha)\right)^\beta}, \quad (\beta > 0) \tag{24}$$

for updating the weights α_i. There is no guarantee that this fixed point will be reached from a particular starting value, such as $\alpha_i = 1/N_c$, through successive applications of (24). Nevertheless, it is clear that (24) will concentrate on the channels with large contributions to the variance, as suggested by stratified sampling. Furthermore, empirical studies show that (24) is successful in practical applications. The value $\beta = 1/2$ has been proposed in [2], but it can be beneficial in some cases to use smaller values like $\beta = 1/4$ to dampen statistical fluctuations.

4 Performance

Both the implementation and the practical use of the algorithm proposed in this note are more involved than the application of the original Vegas algorithm. Therefore it is necessary to investigate whether the additional effort pays off in terms of better performance.

A test version of an implementation of this algorithm, "VAMP", in Fortran [5] has been used for empirical studies. This implementation features other improvements over "Vegas Classic"—most notably system independent and portable support for parallel processing and support for unweighted event generation—and will be published when the documentation [4] is finalized. The preliminary version is available from the author upon request.

4.1 Costs

There are two main sources of additional computational costs: at each sampling point the function $g \circ \phi_i$ has be evaluated, which requires the computation of the $N_c - 1$ maps π_{ij} together with their Jacobians and of the $N_c - 1$ probability distributions g_i of the other Vegas grids (cf. (15)).

The retrieval of the current g_is requires a bisection search in each dimension, i.e. a total of $O((N_c - 1) \cdot n_{\text{dim}} \cdot \log_2(n_{\text{div}}))$ executions of the inner loop of the search. For simple integrands, this can indeed be a few times more costly than the evaluation of the integrand itself.

The computation of the π_{ij} can be costly as well. However, unlike the g_i, this computation can usually be tuned manually. This can be worth the effort if many estimations of similar integrals are to be performed. Empirically, straightforward implementations of the π_{ij} add costs of the same order as the evaluation of the g_i.

Finally, additional iterations are needed for adapting the weights α_i of the multi channel algorithm described in (3). Their cost is negligible, however, because they are usually performed with far fewer sampling points than the final iterations.

4.2 Gains

Even in cases in which the evaluation of g_i increases computation costs by a whole order of magnitude, any reduction of the error by more than a factor of 4 will make the multi channel algorithm economical. In fact, it is easy to construct examples in which the error will be reduced by more than two orders of magnitude. The function

$$f(x) = \frac{b}{144\,\mathrm{atan}(1/2b)} \left(\frac{3\pi\Theta(r_3 < 1)}{r_3^2((r_3 - 1/2)^2 + b^2)} + \frac{2\pi\Theta(r_2 < 1, |x_3| < 1)}{r_2((r_2 - 1/2)^2 + b^2)} \right. \\ \left. + \frac{\Theta(-1 < x_1, x_2, x_3 < 1)}{x_1^2 + b^2} \right), \quad (25)$$

with $r_2 = \sqrt{x_1^2 + x_2^2}$ and $r_3 = \sqrt{x_1^2 + x_2^2 + x_3^2}$, is constructed such that it can easily be normalized

$$\int_{-1}^{1} \mathrm{d}^3x\, f(x) = 1 \quad (26)$$

and allows a check of the result. The three terms factorize in spherical, cylindrical and Cartesian coordinates, respectively, suggesting a three channel approach. After five steps of weight optimization consisting four iterations of 10^5 samples, we have performed three iterations of 10^6 samples with the VAMP multi channel algorithm. Empirically, we found that we can perform four iterations of $5 \cdot 10^5$ samples and three iterations of $5 \cdot 10^6$ samples with the class Vegas algorithm during the same time period. Since the functional form of f is almost as simple as the coordinate transformation, the fivefold increase of computational cost is hardly surprising.

In figure 2, we compare the error estimates derived by the classic Vegas algorithm and by the three channel VAMP algorithm. As one would expect, the multi channel algorithm does not offer any substantial advantages for smooth functions (i. e. $b > 0.01$). Instead, it is penalized by the higher computational costs. On the other hand, the accuracy of the classic Vegas algorithm deteriorates like a power with smaller values of b. At the same time, the multi channel algorithm can adapt itself to the steeper functions, leading to a much slower loss of precision.

The function f in (25) has been constructed as a showcase for the multi channel algorithm, of course. Nevertheless, more complicated realistic examples from particle physics appear to gain about an order of magnitude in

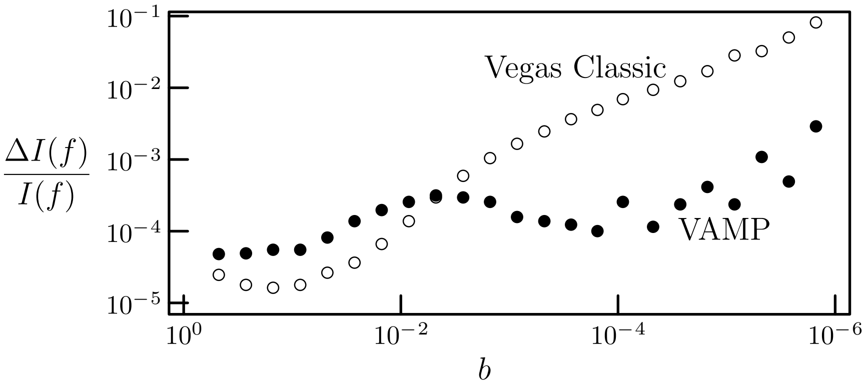

Figure 2: Comparison of the sampling error for the integral of f in (25) as a function of the width parameter b for the two algorithms at comparable computational costs.

accuracy. Furthermore, the new algorithm allows *unweighted* event generation. This is hardly ever possible with the original Vegas implementation, because the remaining fluctuations typically reduce the average weight to very small numbers.

4.3 A Cheaper Alternative

There is an alternative approach that avoids the evaluation of the g_is, sacrificing flexibility. Fixing the g_i at unity, we have for $\tilde{g} : M \to [0, \infty)$

$$\tilde{g} = \sum_{i=1}^{N_c} \alpha_i \left| \frac{\partial \phi_i^{-1}}{\partial p} \right| \tag{27}$$

and the integral becomes

$$I(f) = \sum_{i=1}^{N_c} \alpha_i \int_M \left| \frac{\partial \phi_i^{-1}}{\partial p} \right| \mathrm{d}\mu(p) \, \frac{f(p)}{\tilde{g}(p)} = \sum_{i=1}^{N_c} \alpha_i \int_0^1 \mathrm{d}^n x \, \frac{f(\phi_i(x))}{\tilde{g}(\phi_i(x))} \,. \tag{28}$$

Vegas can now be used to perform adaptive integrations of

$$I_i(f) = \int_0^1 \mathrm{d}^n x \, \frac{f(\phi_i(x))}{\tilde{g}(\phi_i(x))} \tag{29}$$

individually. In some cases it is possible to construct a set of ϕ_i such that $I_i(f)$ can estimated efficiently. The optimization of the weights α_i can again be effected by the multi channel algorithm described in (3).

The disadvantage of this approach is that the optimal ϕ_i will depend sensitively on external parameters and the integration limits. In the approach based on the g in (9) Vegas can take care of the integration limits automatically.

5 Conclusions

We have presented an algorithm for adaptive Monte Carlo integration of functions with non-factorizable singularities. The algorithm shows a significantly better performance for many ill-behaved integrals than Vegas.

The applications of this algorithm are not restricted to particle physics, but a particularly attractive application is provided by automated tools for the calculation of scattering cross sections. While these tools can currently calculate differential cross sections without manual intervention, the phase space integrations still require hand tuning of mappings for importance sampling for each parameter set. The present algorithm can overcome this problem, since it requires to solve the geometrical problem of calculating the maps π_{ij} in (15) for all possible invariants only *once*. The selection and optimization of the channels can then be performed algorithmically.

The application of the algorithms presented here to quasi Monte Carlo integration forms an interesting subject for future research. Other options include maps ϕ_i depending on external parameters, which can be optimized as well. A simple example are rotations, which can align the coordinate systems with the singularities, using correlation matrices [4].

References

[1] G. P. Lepage, J. Comp. Phys. **27**, 192 (1978); G. P. Lepage, Cornell Preprint, CLNS-80/447, March 1980.

[2] R. Kleiss, R. Pittau, Comp. Phys. Comm. **83**, 141 (1994).

[3] R. Kleiss, W. J. Stirling, S. D. Ellis, Comp. Phys. Comm. **40**, 359 (1986); R. Kleiss, W. J. Stirling, Nucl. Phys. **B385**, 413 (1992).

[4] T. Ohl, *`VAMP`, Version 1.0: Vegas AMPlified: Anisotropy, Multi-channel sampling and Parallelization*, Preprint, Darmstadt University of Technology, 1998 (in preparation).

[5] International Standards Organization, *ISO/IEC 1539:1997, Information technology — Programming Languages — Fortran,* Geneva, 1997.