КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ФИЗИКЕ

THE BOUNDARY LAYER PROBLEM IN BAYESIAN ADAPTIVE QUADRATURE

Gh. Adam, S. Adam

Joint Institute for Nuclear Research, Dubna

Horia Hulubei National Institute for Physics and Nuclear Engineering (IFIN-HH),
Magurele–Bucharest, Romania

The boundary layer of a finite domain $[a, b]$ covers mesoscopic lateral neighbourhoods, inside $[a, b]$, of the endpoints a and b. The correct diagnostic of the integrand behaviour at a and b, based on its sampling inside the boundary layer, is the first from a set of hierarchically ordered criteria allowing a priori Bayesian inference on efficient mesh generation in automatic adaptive quadrature.

Пограничный слой ограниченного отрезка $[a, b]$ покрывает мезоскопические боковые области внутри отрезка с концами a и b. Корректная диагностика поведения подынтегрального уравнения в a и b, основанная на его выборке внутри пограничного слоя, является первой из набора иерархически упорядоченных критериев, допускающих a priori байесовский вывод об эффективной сеточной генерации в автоматической адаптивной квадратуре.

PACS: 02.60.Jh; 02.50.Tt; 02.60.Ph

INTRODUCTION

The boundary layer problem in numerical quadrature is the first, and probably the most difficult to solve, from a Bayesian chain aiming either to implement the automatic mesh generation by the adaptive quadrature methods under sound expectation of *reliable local quadrature rule* $\{q, e\}$ outputs, or to provide early detection of the origins of code failure.

The need of such a stringent requirement, which goes well beyond the usual implementation of the automatic adaptive quadrature rules (see, e.g., [1–3]) arises in the numerical exploration of the predictions of models describing phase transitions in complex physical systems, which critically depend on the realization of (unknown in advance) values of some specific parameters (like, e.g., the hole or electron doping level in high-T_c superconducting materials [4]). The solution of the resulting parametric integrals makes use of the existing library programs the reliability of the outputs of which is heavily based on user's ability to choose the suitable procedure from a proposed menu. The impossibility to know in advance the detailed behaviour of the integrand function over the whole class of parametric integrals forces the use of a trial and error approach which may result in unnoticed unreliable $\{q, e\}$ pairs and, consequently, bad output failure.

The a posteriori assessment of the reliability of the $\{q, e\}$ pair over the current integration subrange [5,6] solves only half of the problem since the code remains highly inefficient. The a priori verification of the conditioning of the integrand profile at the quadrature knots, which

was proposed by us some time ago [7], needed, in fact, a whole set of hierarchically ordered criteria providing *Bayesian inference* [8] on the status of the *gradually generated* integrand profile at newly added quadrature knots.

The *root* of the resulting Bayesian inference decision tree is the *diagnostics of the behaviour of the integrand function $f(x)$ at the boundaries a and b of the finite integration domain* $[a, b]$. To set it, a suitable integrand sampling is required inside a *mesoscopic neighbourhood of the boundary layer of* $[a, b]$.

The present paper generalizes the analysis done in [9] over minimal four-point partitions inside the mesoscopic regions associated to each of the ends a and b of $[a, b]$. By allowing an *unrestricted* four-point partition, the diagnostic failures stemming from the inadequacy of a frozen integrand sampling are avoided, resulting in analysis reliability enhancement for difficult integrand functions.

1. DIAGNOSTICS AND BAYESIAN INFERENCES FROM THE BOUNDARY LAYER ANALYSIS

Let x_r denote either $fl(a)$ or $fl(b)$, the floating point representations of the endpoints a and b of $[a, b]$. The analysis establishes the status of the value $f_{x_r} = fl(f(x_r))$, the *floating point representation of the computed value of the integrand*, $f(x_r) \in \mathbb{R}$, as follows:

(i) *Diagnostics: Smooth integrand behaviour.*

Bayesian inference: *Regular* quadrature knot mesh over $[a, b]$ can be generated starting from the x_r endpoint.

Supplementary information: The analysis also generates an estimate of the lateral derivative $f'(x_r)$ inside $[a, b]$. This will serve to the formulation of acceptance check criteria for the two quadrature knots which lie nearest and next nearest to x_r. In case of rejection, the decision to perform the *immediate subrange subdivision of* $[a, b]$, without generating the integrand profile at the remaining quadrature knots, is taken.

(ii) *Diagnostics: Endpoint or outer singularity of $f(x)$ or its derivatives.*

Bayesian inference: *Slow convergence* is to be expected under subrange subdivision. Use of a specific subrange subdivision procedure based on bisection together with a convergence acceleration procedure (e.g., the epsilon extrapolation algorithm) are a must.

(iii) *Diagnostics: Inner nearby singularity.*

Bayesian inference: The occurrence of an offending inner singular point x_0 near x_r is to be further confirmed. Under affirmative diagnostics, x_0 is to be located to machine accuracy. The input integral is then split into two integrals, over $[a, x_0)$ and $(x_0, b]$, each being further processed following the procedure adequate for endpoint singularity.

(iv) *Diagnostics: Inner nearby finite jump.*

Bayesian inference: Further confirmation of the existence of an offending inner jump point x_0 near x_r is necessary. Under affirmative diagnostics, x_0 is to be located to the machine accuracy. The input integral is then split into two integrals, over $[a, x_0^-)$ and $(x_0^+, b]$, with $f(x_0^\mp)$ and $f'(x_0^\mp)$ taking values equal to the lateral limits of $f(x)$ and $f'(x)$ at x_0. The resulting integrals over the two subintervals are to be solved for smooth $f(x)$.

(v) *Diagnostics: Irregular behaviour.*

Bayesian inference: The output of the automatic procedure could hardly be taken for reliable. Clarification of the offending integrand behaviour is a must.

(vi) *Diagnostics: Smooth integrand behaviour at both ends a and b.*

Bayesian inference: Early check for the presence of an *oscillatory or odd integrand* is useful. If an affirmative diagnostics is issued, then *define a ceiling accuracy of the expected output*, originating in severe precision loss due to heavy cancellation by subtraction.

2. UNRESTRICTED LEAST SQUARES ANALYSIS

Theorem 1. The function $f : [a, b] \subset \mathbb{R} \to \mathbb{R}, f = f(x)$, is *smooth* inside a lateral mesoscopic neighbourhood $V(x_r) \subseteq [a, b]$ of the reference abscissa x_r denoting the floating point representation of either the end a or the end b of $[a, b]$, provided the computed values of the first-order divided differences of $f(x)$ over any abscissa sampling inside $V(x_r)$ are *independent of the choice of the sampling abscissas.*

Remark 1. The result stated in Theorem 2 essentially follows from the property that, for any reference abscissa $x_r \in D \subset \mathbb{R}$ of a continuous twice differentiable function $f : D \to \mathbb{R}, f = f(x)$, a nonvanishing neighbourhood $V(x_r) \subseteq D$ does exist inside which the *linear* Taylor series expansion of $f(x)$ around $f(x_r)$ holds true within some predefined accuracy threshold $0 < \varepsilon \ll 1$.

The numerical check of the continuity of $f(x)$ at the ends of the integration domain $[a, b]$ can only be done from a *sampling of its computed values*, $\{f_i = fl(f(x_i)) | i = 0, 1, \ldots, m\}$, over a set of $m + 1$ machine number arguments $S_m(x_r) = \{x_i \in V(x_r) | i = 0, 1, \ldots, m\}$, $x_r \in S_m(x_r)$, $m \geqslant 3$. If $\{f(x_i)) | i = 0, 1, \ldots, m\}$ denote the *actual* values of $f(x_i)$ over $S_m(x_r)$, then, due to the round-off, $f(x_i) - f_i \neq 0$ in general. As a consequence, the best information on the smoothness properties of $f(x)$ at x_r following from the set $\{x_i, f_i\}$ is obtained from the scrutiny of the properties of a second degree polynomial least squares fit to the floating point data.

A problem in terms of *machine number abscissas* is obtained by the scale transformation

$$x_i = x_0 + \xi_i h_r, \quad i = 0, 1, \ldots, m; \quad \xi_i \in \mathbb{Z}; \quad \xi_0 = 0, \tag{1}$$

where h_r denotes the algebraic distance from x_r to its nearest machine number inside $[a, b]$. This leads to the second degree fitting polynomial

$$y_2(x_i) = \alpha_0 \pi_0(\xi_i) + \alpha_1 \pi_1(\xi_i) + \alpha_2 \pi_2(\xi_i), \tag{2}$$

spanned by the *orthonormal* basis polynomials $\pi_k(\xi_i)$, $k = 0, 1, 2$.

A *linear* polynomial fitting over $S_m(x_r)$ is obtained provided $\alpha_2 \pi_2(\xi)$ is *negligible everywhere* at $\xi \in [\xi_{\min}, \xi_{\max}]$, $\xi_{\min} = \min\{\xi_0, \xi_1, \ldots, \xi_m\}$, $\xi_{\max} = \max\{\xi_0, \xi_1, \ldots, \xi_m\}$. Making all the calculations requested by the least squares procedure then results in the statement of the theorem, QED.

Corollary 1. *If we assume a minimal mesh sampling $S_3(x_r)$ characterized by the abscissa set $\xi_0 = 0$, $\xi_1 = p > 0$, $\xi_2 = \mu p > \xi_1$, $\xi_3 = q$, $|\xi_3| \ll \xi_1$, then the following smoothing criteria emerge*:

$$d_{20} - \mu d_{10} \approx 0, \quad \mu d_{30} - \frac{q}{p} d_{20} \approx 0, \quad \frac{q}{p}\left[\frac{q}{p} d_{10} - d_{30}\right] \approx 0, \quad d_{ij} = f_i - f_j. \tag{3}$$

Corollary 2. *If we assume a minimal mesh sampling $S_3(x_r)$ characterized by the abscissa set $\xi_0 = 0$, $\xi_1 = p \sim \xi_2 = q$, $\xi_3 = r$, $\max\{\xi_1, \xi_2\} \ll \xi_3$, then the following smoothing criteria emerge*:

$$pd_{20} - qd_{10} \approx 0, \quad qd_{30} - rd_{20} \approx 0, \quad rd_{10} - pd_{30} \approx 0. \tag{4}$$

Corollary 3. *If the analysis issued the diagnostics of smooth $f(x)$ at the endpoint x_r, then the following estimate for the lateral first-order derivative $f'(x_r)$ inside $[a, b]$ holds,*

$$f'(x_r) \approx \frac{1}{(m+1)\bar{\delta}^2 h_r} \sum_{i=1}^{m} \delta_i d_{i0}, \tag{5}$$

$$\delta_i = \xi_i - \bar{\xi}, \quad \bar{\xi} = \frac{1}{m+1} \sum_{i=0}^{m} \xi_i, \quad \bar{\delta}^2 = \frac{1}{m+1} \sum_{i=0}^{m} \delta_i^2. \tag{6}$$

3. DIAGNOSTIC UNIQUENESS

To allow useful inferences, the analysis done in the previous section has to be supplemented, on one side, with a quantitative measure of the smallness of the differences defined in Corollaries 1 and 2 and, on the other side, with qualitative criteria able to single out those specific unique features which characterize other integrand behaviours inside a mesoscopic neighbourhood of x_r.

3.1. Smoothness Threshold. The practical implementation of the smoothness criteria (3) and (4) compares the magnitudes of the expressions entering the left-hand sides of these equations with an *integrand-dependent upper threshold*,

$$0 < \tau_f = \tau_0 \varepsilon_0 \max\{|f_0|, |f_1|, |f_2|, |f_3|\}, \tag{7}$$

where ε_0, the machine epsilon with respect to addition, defines the machine accuracy dependent parameter of the solution, while τ_0, $0 < \tau_0 \ll \varepsilon_0^{-1/2}$, is a heuristic parameter intended to overcome possible diagnostic errors coming from the normal roundoff noise. In practice, we have found that a value $\tau_0 = 2^{10}$ resulted in adequate diagnostics for all the tested smooth case study functions $f(x)$.

The diagnostic reliability decreases as long as the magnitudes of one or more of the quantities entering Eqs. (3) or (4) get near τ_f.

3.2. Endpoint or Outer Singularity. The basic features of such a behaviour are: inward monotonically decreasing $|f(x)|$ under $d_{i,i-1}d_{i+1,i} > 0$; steep variation of $f(x)$ over argument distances separated by one or a few machine numbers; inward monotonically decreasing $|f'(x)|$.

If the sampling of Corollary 1 is generated, then the addition of two supplementary abscissas at $(\xi_1 + \xi_3)/2$ and $(\xi_1 + \xi_2)/2$, respectively and the confirmation of the above-mentioned three features once again over the resulting smaller subranges distinguishes singular behaviour from an inner finite jump near x_r.

3.3. Inner Nearby Singularity. The features of this kind of behaviour are mirror reflected with respect to those of an endpoint or outer singularity.

3.4. Inner Nearby Finite Jump. There are two hints suggesting this diagnostics. First, the sharp increase of the magnitude of one of the first-order divided differences approximating $f'(x)$ over the subranges defined by sampling. Second, the feature gets enhanced if a finer partition is defined over the subrange in question.

3.5. Irregular Behaviour. This negative diagnostics is usually associated with the occurrence of rapid oscillations of $f(x)$ which make the usual local quadrature rules ineffective.

4. CODE ROBUSTNESS: HARDWARE AND SOFTWARE ENVIRONMENT

The analysis described above yields diagnostics issued by a code which runs within an environment defined by the hardware and software at hand. The *code robustness, reliability, and portability* are secured provided several delicate points are adequately solved.

The last code versions were run on several PCs with Intel 4+, AMD32, or AMD64 processors, Linux 2.4 or Linux 2.6+ operating systems, and the GNU gcc compiler incorporating Fortran 77. Early code versions were also run on SUN workstations at BLTP–JINR, or under Microsoft XP OS.

The study of the conformity of the hardware and software of the above-mentioned systems to the *IEEE 754 standard* which governs the floating point arithmetic (see, e.g., [10, 11]), revealed the occurrence of four instances where the requirements of the standard were more or less frequently infringed: (i) *length of the significand;* (ii) *floating point comparisons;* (iii) *code optimization;* (iv) *underflow threshold.*

4.1. Length of the Floating Point Double Precision Significand. Under the standard compliant value of the machine epsilon with respect to addition, $\varepsilon_0 = 2^{-53}$, often analysis failure has been noticed. This was identified to stem from the *loss of the last, assumed significant, 53rd bit* under reuse of *exact* data after their RAM/cache storage.

The simplest possible case study of hidden bit loss is illustrated in Figure: the quantities $a_k = 1 + k\varepsilon_0$, $\varepsilon_0 = 2^{-53}$, $k = -20(1)20$, are computed and stored in an array (to force RAM/cache storage) and then the differences $\delta_{k+1,k} = a_{k+1} - a_k$ are plotted. Under IEEE 754 standard observance, the *constant answer*, $\delta_{k+1,k} = \varepsilon_0$ should have been obtained. However, this result is obtained indeed under $\varepsilon_0 = 2^{-52}$.

Figure, b illustrates the same effect when exploring integrand behaviour around a singular point on an example taken from QUADPACK, [1, p.110],
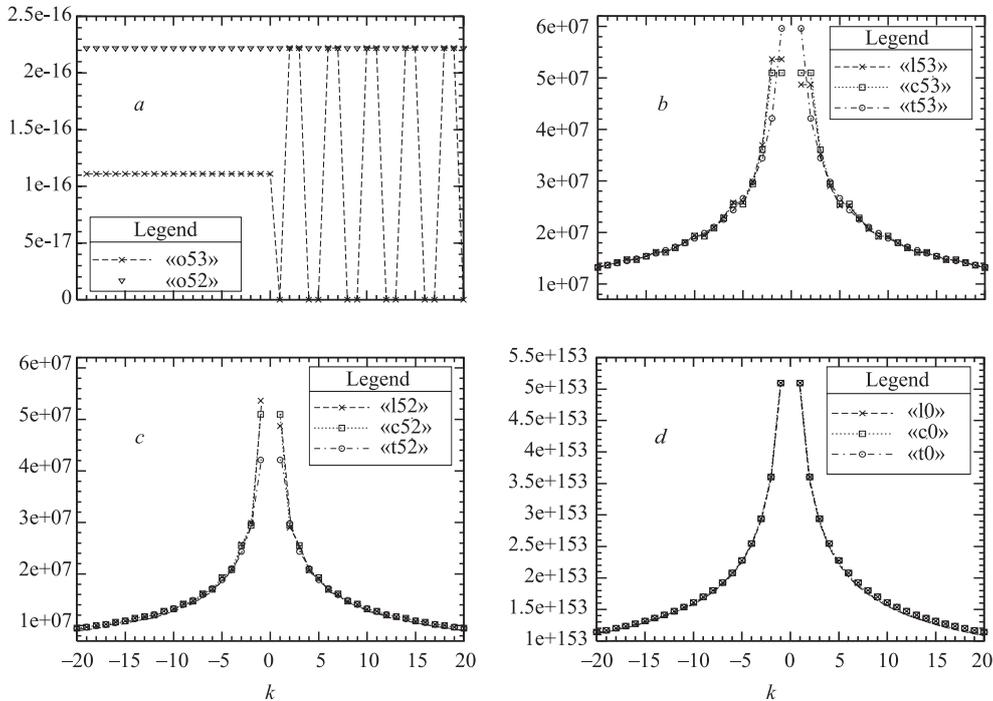
$$f(x) = \frac{1}{|x^2 + 2x - 2|^{1/2}} = \frac{1}{\sqrt{|(x - x_p)(x - x_m)|}},$$

where

$$x_p = \sqrt{3} - 1, \ x_m = -\sqrt{3} - 1. \tag{8}$$

The values $\{f(x_k); \ x_k = fl(x_p)(1 + k\varepsilon_0); \ \varepsilon_0 = 2^{-53} | \ k = -20(1)20\}$ have been computed and compared with the exact ones. Several spuriously equal output pair are noticed at pairs of assumed neighbouring machine number arguments, which spoil the reliability of the analysis.

Increase of ε_0 to $\varepsilon_0 = 2^{-52}$ rules out such spurious pairs (Figure, c).

Influence of the machine epsilon definition on output reliability. *a*) Check of the fulfilment of the reliability criterion $fl(1 + (k + 1)\varepsilon_0) - fl(1 + k\varepsilon_0) = \varepsilon_0$. (Data set «o53» shows that our system is not compliant to the IEEE 754 standard value $\varepsilon_0 = 2^{-53}$; data set «o52» shows that $\varepsilon_0 = 2^{-52}$ is the right choice.) *b–d*) Comparison of two code outputs with the true results for the case study function (8): *b*) spuriously equal function pair values occur under $\varepsilon_0 = 2^{-53}$ at arguments intended to represent neighbouring machine numbers; *c*) the spuriously equal pairs disappear, but heavy precision loss and output precision path dependence are still present near the singular point $x_p = \sqrt{3} - 1$ under $\varepsilon_0 = 2^{-52}$; *d*) all difficulties are solved if singularity is moved to the origin, where $\varepsilon_0 = u = 2^{-1023}$. (Legend prefixes: «t» — *true* results; «c» — *unique* floating point value of x_p used inside all procedures; «l» — calculation of $f(x)$ is done using a *locally processor produced approximation* for x_p, which is more accurate than that transferred in-between other procedures)

4.2. Floating Point Comparisons. In the example (8), $fl(x_p)$ may be either transferred from a procedure to another one using the $fl_{52}(x_p)$ value which retains the most significant 52 binary bits in the significand, or it may be directly computed in CPU from the original expression $x_p = \sqrt{3} - 1$, in which case the processor stored $fl_{64}(x_p)$ value retains the most significant 64 binary bits. The IEEE 754 standard asks that $fl_{64}(x_p).\texttt{EQ.}fl_{52}(x_p) = .\texttt{TRUE.}$ None of the two compilers available to us (f77, C++ gcc) did obey to this requirement.

Since the use of floating point comparison was unavoidable, special care was taken to exclude all the possible fl_{52} to fl_{64} comparisons.

4.3. Code Optimization by the Compiler. We might try to write a code in which use of fl_{52} values is always secured in comparison operations by asking the transfer of each variable entering such an operation to RAM/cache, followed by their transfer back to CPU. In

its search for «efficiency» increase, the code optimization by the compiler finds such a trick «unnecessary», thus spoiling the output correctness. The observation is not singular [12].

4.4. The Underflow Threshold. On all but one computer at our disposal, the standard value $u = 2^{-1023}$ was found to be right. However, on one of the mentioned SUN workstations the code crashed at $u = 2^{-1023}$, while the value $u = 2^{-1022}$ was found to be OK.

4.5. Catastrophic Precision Loss in the Neighbourhood of a Non-zero Singular Point. Two supplementary features present in Figure b, c deserve consideration: (i) The computed function values are different from the exact ones *starting with the most significant bit*. (ii) The use of the primary QUADPACK expression listed in Eq. (8) (data labelled «l53», «l52») infringes the left-right symmetry of the exact data.

In spite of this severe precision loss due to cancellation by subtraction, correct inferences based on the use of the unique features characterizing a singular behaviour (Subsec. 3.2) are still possible due to the fact that the computed «wrong data» *preserve* the ordering relationships characteristic to the «true data».

If a substitution of variable which moves the singularity to the origin is possible, then all the difficulties enumerated at Subsecs. 4.2, 4.3, and 4.5 are completely removed (Figure, d).

CONCLUSIONS

The boundary layer problem asks for accurate and reproducible diagnostics of integrand $f(x)$ behaviour at the endpoints a and b of a finite integration domain $[a, b]$.

In the present paper we have discussed several critical issues which dramatically influence code robustness and reliability. Generalizations of the results previously reported in [9] allows significant improvement of code quality by:

(i) Definition of smooth behaviour from an *unrestricted* least squares analysis over small (mesoscopic) neighbourhoods of the endpoints a and b. This secures the derivation of continuity criteria valid *everywhere* over the mesoscopic range where the analysis is done.

(ii) Formulation of qualitative diagnostic criteria which reliably single out the various kinds of integrand behaviour even under severely damaged accuracy of the computed data.

(iii) Derivation of accurate tests for reliable definition of the machine epsilon with respect to the addition.

(iv) Identification of the critical hardware and software features which could spoil the correctness of the diagnostics by deviation from the IEEE 754 standard and code reformulation such as to become insensitive to such a drawback of the computing environment.

The correct solution of the boundary layer problem is the first from a set of hierarchically ordered problems the solutions of which should allow a priori Bayesian inferences on efficient and reliable mesh generation within automatic adaptive quadrature. The solution of such problems is planned to be discussed in subsequent reports.

REFERENCES

1. *Piessens R. et al.* QUADPACK, a Subroutine Package for Automatic Integration. Berlin: Springer Verlag, 1983.

2. *Davis P. J., Rabinowitz P.* Methods of Numerical Integration. 2nd ed. Orlando (Fla), USA: Acad. Press, 1984.

3. *Krommer A. R., Ueberhuber C. W.* Computational Integration. Philadelphia: SIAM, 1998.

4. *Plakida N. M.* High-Temperature Superconductivity: Experiment and Theory. Berlin: Springer-Verlag, 1995.

5. *Adam Gh., Adam S.* // Comp. Phys. Commun. 2001. V. 135. P. 261.

6. *Adam Gh., Adam S., Plakida N. M.* // Comp. Phys. Commun. 2003. V. 154. P. 49.

7. *Adam Gh., Adam S.* // ICCAM2004 Conf., Univ. Leuven, Belgium, July 25–31, 2004.

8. *Jaynes E. T.* Bayesian Methods: An Introductory Tutorial // Maximum Entropy and Bayesian Methods in Applied Statistics / Ed. J. H. Justice. Cambridge: Cambridge Univ. Press, 1986.

9. *Adam Gh. et al.* // Rom. Rep. Phys. 2006. V. 58. P. 107.

10. *Goldberg D.* What Every Computer Scientist Should Know about Floating-point Arithmetic // Comp. Surveys. 1991; available at web site http://www.validlab.com/goldberg/paper.pdf

11. *Kahan W.* Lecture Notes on the Status of IEEE Standard 754 for Binary Floating-Point Arithmetic. 1996; available at web site http://www.cs.berkeley.edu/˜wkahan/ieee754status/ieee754.ps

12. *Kahan W.* How Futile Are Mindless Assessments of Roundoff in Floating-Point Computation (in progress); available at the web site http://www.cs.berkeley.edu/˜wkahan/Mindless.pdf