Russian Academy of Science, Siberian Branch
BUDKER INSTITUTE OF NUCLEAR PHYSICS

S.V. Miginsky

NEW QUADRATURES WITH LOCAL ERROR ESTIMATION AND TWO STRATEGIES OF STEP CONTROL IN CALCULATION OF DEFINITE INTEGRALS

Budker INP 2001-18

Novosibirsk
2001

New Quadratures with Local Error Estimation and Two Strategies of Step Control in Calculation of Definite Integrals

S.V.Miginsky

Budker Institute of Nuclear Physics, 630090 Novosibirsk, Russia
S.V.Miginsky@inp.nsk.su

Abstract

Four new quadratures with local error estimation were proposed and two of them were numerically tested. Their effectiveness and robustness seem to be higher in comparison to the well known Gauss-Kronrod one. Also two strategies of step control were considered. A combination step control algorithm adapted to the proposed quadratures was implemented and tested. In addition, several other aspects of numerical integration were discussed.

Новые квадратуры с оценкой локальной погрешности и две стратегии регулировки шага при вычислении определенных интегралов

С.В.Мигинский

Институт ядерной физики им. Г.И. Будкера СО РАН
630090, Новосибирск, Россия

Аннотация

Предложены четыре новые квадратуры с оценкой локальной погрешности, две из них численно тестированы. Их эффективность и надежность оказались выше, чем у известной квадратуры Гаусса-Кронрода. Рассмотрены также две стратегии регулировки шага интегрирования. В тестированных интеграторах применен комбинированный алгоритм регулировки шага, адаптированный к предложенным квадратурам. Кроме того, обсуждается ряд других аспектов численного интегрирования.

1. Introduction

To minimize the expense of definite integral calculation, the length of each integration step should be chosen so that the local errors are equal. Thus, a non-expensive and flexible method of local error estimation should be used to make the procedure effective enough. One known method is the Richardson extrapolation [1], [2]. According to this method, two consequent equal steps are made and after that they are covered by a double-length step using the same formula of the degree of exactness k. Then, if the results are y and y_d correspondingly, one can estimate the local error

(1) $$\Delta y \sim \frac{y_d - y}{2^k - 1}$$

and get the degree of exactness $k+1$ result

(2) $$\tilde{y} \sim y + \frac{y - y_d}{2^k - 1}.$$

The degree of exactness k means here, that if the integrated function is smooth enough,

(3) $$\lim_{h \to 0} R_k / h^{k+1} = const,$$

where R_k is the local error and h is the integration step. Thus the global error is limited by $const \cdot h^k$.

Another way is to construct an embedded pair of quadratures of different order using the same nodes. Then one can use its difference as a local error estimation. Probably the most effective known quadrature of this type is Gauss-Kronrod one [3]-[6] and references there. The Gauss-Kronrod quadrature based on the Gaussian one having n nodes and $k = 2n$ degree of exactness contains $n+1$ additional nodes. Its degree of exactness is $k = 2\left[(3n+3)/2\right]$.

One more quadrature derived using similar assumptions is the anti-Gaussian one [7]. It is based on the Gaussian quadrature having n nodes, contains $n+1$ ones, the degree of exactness $2n$ and the error constant, equal to Gaussian one, but of the opposite sign. Then local error estimation is a half of the difference between these quadratures.

In all the mentioned cases additional evaluations of the integrated function are necessary to perform local error estimation. Several attempts to construct more effective embedded quadratures are discussed below.

An algorithm of step control also affects the total effectiveness of an integrating code. At least two groups of step control strategies can be used in such code: consequent and binary-tree like ones. In the first case the current integration step is admitted or rejected depending on whether the estimated local error is satisfactory or not. Then the length of the next step or of the decreased one is predicted using the estimated error. According to another strategy the rejected step is divided by two equal ones and the procedure is repeated for each half and so on. Both strategies and their combinations are also discussed further.

2. Embedded quadratures based on Gaussian ones

Let's try to construct more effective embedded quadrature formulae. In [3] A.S.Kronrod proposed a definition of the optimal embedded pair, that provides the maximum order of the lower order quadrature, and the maximum order of the higher-order one if the previous value is the same, for a given total number of nodes. He found these pairs called now Gauss-Kronrod ones for arbitrary odd number of nodes. He proved also the theorem, that at least $n+1$ nodes is necessary to get the order n of the lower order quadrature in a pair. It should be stressed that the number of nodes is not a good measure of the effectiveness of a quadrature. It seems to be reasonable to measure the effectiveness by the number of integrated function evaluations per step, all factors being equal. For example, the number of function estimations is less by one than the number of nodes for quadratures having the nodes at the both borders of the step, as the function values at these nodes are used simultaneously by two adjacent steps. This takes place, say, for the Lobatto quadratures and the Newton-Cotes ones.

Now let's construct embedded pairs based on the Gaussian quadratures [8], 25.4.29 not only by adding some nodes, but by excluding some nodes and after that adding the nodes ± 1. If one takes the Gaussian quadrature having n nodes and the order $2n$, excludes two symmetric nodes or the central one (if any), and adds the two ones ± 1, he obtains a new quadrature having n nodes ($n+1$, if the central node is excluded) and the order $k = 2[(n+1)/2]$. The question is which nodes should be excluded. If the border nodes are excluded from the base Gaussin quadrature, the error

constant of the obtained quadrature gets the minimum. I don't know a proof of this fact, I only checked it for n up to 9. The coefficients of the Gauss based quadratures are presented in Appendix I. The error constant c_k means here the coefficient before the appropriate derivative in the remainder for the interval (-1, 1):

(4) $$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} a_i f(x_i) + R_k ,$$

(5) $$R_k = c_k f^{(k)}(\xi), \ (-1 < \xi < 1) .$$

Thus, taking n nodes, one can construct the embedded pair of the orders $2n-4$ and $2[(n-1)/2]$. Note that the proposed pairs are better than the Gauss-Kronrod ones for arbitrary odd $n \geq 15$ according to Kronrod's criterion. If consequent steps are used for calculation and the criterion mentioned above is applied, the proposed pairs are better than the Gauss-Kronrod ones for any number of function calculation per step $n = 4,\ 6,\ \geq 8$.

3. Embedded quadratures based on Lobatto ones

Now let's consider the Lobatto quadrature as the base [8], 25.4.32. Compared to the Gaussian one, this quadrature needs one more node to get the same degree of exactness. Nevertheless, the number of integrated function evaluations for one step is the same as each border value is used by two subsequent steps. Also the remainder of the Lobatto quadrature is larger by factor $(n+1)/n$, where n is the number of nodes of the Gaussian one. But this factor is almost 1.

3.1. Lobatto-Kronrod quadrature

First of all, consider a quadrature that can be called Lobatto-Kronrod one. It contains $n+1$ nodes of the Lobatto quadrature and n additional nodes so that its degree of exactness is $k = 2([3n/2] + 1)$. One can find these additional nodes using the same procedure as described in [6]. The difference is the polynomials corresponding to the Lobatto quadratures are the first derivatives of the Legendre polynomials $P_n'(x)$ and the weight function is $w(x) = x^2 - 1$. Thus, for the $n+1$-nodes Lobatto quadrature one should take the $n-1$-order polynomial $P_n'(x)$ and construct the appropriate Stieltjes polynomial $E_n(x)$ using the following set of linear equations:

(6) $$\int_{-1}^{1}(x^2-1)P_n'(x)E_n(x)x^m dx\begin{cases}=0, & 0\le m<n\\ \neq 0, & m=n\end{cases}.$$

The roots of $E_n(x)$ are the additional nodes of the Lobatto-Kronrod quadrature. The nodes and coefficients of these quadratures are placed in Appendix II. Note that even n is preferable due to the integer part of $3n/2$ in the formula for the degree of exactness. The main advantage of the proposed quadrature is that it uses one less function estimation per step compared to the Gauss-Kronrod one of the same order of the lower-order quadrature. According to Kronrod's definition this quadrature is worse than Gauss-Kronrod one.

3.2. Lobatto based quadrature

Another way to construct an embedded quadrature for error estimation is to remove two symmetric nodes (or the central node, if exists) from the Lobatto one. Then, if one took the Lobatto quadrature with n+1 nodes and $k = 2n$ degree of exactness, the embedded one is of $k = 2[n/2]$ degree of exactness. The key advantage of the Lobatto quadrature in this case is one more node in comparison to Gaussian one. It permits to get higher order of the embedded quadrature. Note that one should use even n. In this case the degree of exactness of the lower order quadrature is n, while in the other case n - 1. Another question is which pair of nodes or the central node should be removed. The quadrature without the border nodes possesses the best error constant. I don't have proof of this fact for arbitrary n, but I checked it for n up to 10. Coefficients of the Lobatto based embedded quadratures see in Appendix III. Comparison of necessary numbers of function evaluations and achieved degrees of exactness are placed in Table 1.

Table 1.

Function eval. per step	Degree of exactness of result				Degree of exactness of error estimation
	Gauss-Kronrod	Gauss based	Lobatto-Kronrod	Lobatto based	
n	$2\left\lfloor\frac{3n+3}{4}\right\rfloor$ (odd only)	$2n-2$	$2\left\lfloor\frac{3n+4}{4}\right\rfloor$ (even only)	$2n$	$2\left\lfloor\frac{n}{2}\right\rfloor$
2	-	2	4	4	2
3	6	4	-	6	2
4	-	6	8	8	4
5	8	8	-	10	4
6	-	10	10	12	6
7	12	12	-	14	6
8	-	14	14	16	8
9	14	16	-	18	8
10	-	18	16	20	10
11	18	20	-	22	10

Note that the Lobatto based quadrature ever uses less function evaluations or provides better degree of exactness or both simultaneously in each case except of $n = 3$. Another advantage of the Lobatto based quadrature is the simple and obvious way to construct it. Its nodes are the same as for the Lobatto one. It should be stressed, that even according Kronrod's criterion, the Lobatto based quadrature is better than the Gauss-Kronrod one for any number of nodes $n \geq 5$ (the degree of exactness of the higher order quadrature is greater or equal and the error constant of the lower order one is less). The error constants of mentioned quadratures are presented in Table 2.

Table 2.

Degree of exactness	Error constants c_k (absolute values)					
	Gaussian	Gauss-Kronrod	Gauss based	Lobatto	Lobatto-Kronrod	Lobatto based
2	0.333	-	0.667	0.667	-	0.333
4	0.00741	-	0.0111	0.0111	0.0111	0.00476
6	$6.35\cdot10^{-5}$	$6.35\cdot10^{-5}$	$5.13\cdot10^{-5}$	$8.47\cdot10^{-5}$	-	$3.21\cdot10^{-5}$
8	$2.88\cdot10^{-7}$	$9.00\cdot10^{-8}$	$1.48\cdot10^{-7}$	$3.60\cdot10^{-7}$	$3.60\cdot10^{-7}$	$1.23\cdot10^{-7}$
10	$8.08\cdot10^{-10}$	-	$2.93\cdot10^{-10}$	$9.70\cdot10^{-10}$	$1.70\cdot10^{-10}$	$3.05\cdot10^{-10}$
12	$1.54\cdot10^{-12}$	$5.86\cdot10^{-13}$		$1.80\cdot10^{-12}$	-	
14	$2.13\cdot10^{-15}$	$1.15\cdot10^{-16}$		$2.43\cdot10^{-15}$	$6.16\cdot10^{-16}$	
16	$2.22\cdot10^{-18}$	-		$2.50\cdot10^{-18}$	$2.03\cdot10^{-19}$	
18	$1.82\cdot10^{-21}$	$1.38\cdot10^{-22}$		$2.03\cdot10^{-21}$	-	
20	$1.20\cdot10^{-24}$	-		$1.32\cdot10^{-24}$		

3.3. Error estimation using nodes of the previous step

One can also estimate the local error using nodes from the previous step. In this case one more node in the Lobatto quadrature is the key advantage. The total number of nodes in two steps of the Gaussian quadrature of order $2n$ is $2n$, so it is impossible to construct a higher order quadrature using them. In the case of the Lobatto quadrature the number of nodes is $2n+1$, so one can construct a $2n+1$-order quadrature using the same nodes. Its coefficients see in Appendix IV. One can use any of the mentioned embedded quadratures as the integral evaluation and their difference as the local error estimation. The error constant of the proposed quadrature is the same of the Lobatto one, but has the opposite sign. The main advantage of this pair is that it has maximum possible degree of exactness for a given number of function evaluations per step. It also evaluates the local error well as the orders of the quadratures in the pair differ by one. The main its drawback is that the higher order quadrature is not local, so ξ in (5) can vary in limits (-3, 1), error estimation looks not so reliable, and some starting procedure is necessary as there is no previous step for the first step of integration. Also step control in an integrator using this pair can't be so flexible as in the case of a pure one-step quadrature. It is necessary to make at least two equal steps to evaluate the local error. Of course, one can calculate dynamically such quadrature for unequal steps, but overhead expenses in this case increase dramatically.

It is possible to bring this idea ad absurdum. An arbitrary order quadrature can be constructed using an appropriate number of previous nodes and only one new function evaluation. Unfortunately the error constant of such formula is much larger than one of the Gaussian quadrature normalized to one node and the same interval. So one can't decrease the number of function evaluations this way, but only get fantastic overhead expenses. On the contrary, the proposed pair uses the pure one-step Lobatto quadrature for integral evaluation and the non-local two-step quadrature for local error estimation.

4. Two strategies of step control

It should be stressed that the maximum integration step $h_{\max}$ together with the maximum value of local error usually called Tol are to be specified before integrating. If the first value is omitted, that is taken equal to the whole integration interval, and one would like to calculate something like this

(7) $$\int_0^{10} \sin^{100} x \; dx \,,$$

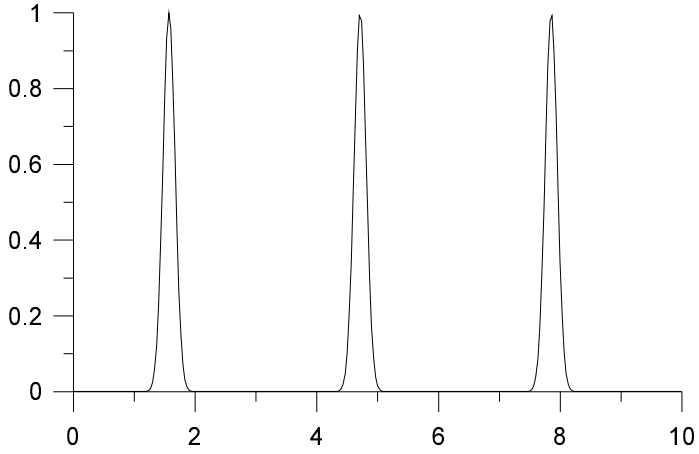

Fig. 1. $\sin^{100} x$.

the result probably will be $\cong 0$ instead of $\cong 0.750123$. It takes place if the nodes of the quadrature lay in the area where the function is $\cong 0$, so the estimated local error is satisfactory. One should specify the maximum distance between the nodes ≈ 0.5 in this case to ensure the valid result.

As mentioned above, at least two strategies of step control are possible. Note that only ones optimizing local error are discussed. According to the first strategy, that can be called "subsequent", the current step is admitted if the estimated local error $R_k \le Tol$. In the other case the current step is rejected and should be decreased. The length of the next step (if admitted) or of the decreased one (if rejected) is usually estimated as

(8) $$h_+ = A \cdot h \cdot \left(B \cdot Tol \,/\, R_k \right)^{1/(k+1)},$$

where h is the current step length, k is the degree of exactness of the error estimator, A and B are warranty factors, both ≤ 1. The ratio h_+ / h should be limited to avoid too great step increase if the estimated error $\to 0$.

Another strategy, that can be called "tree-like", assumes that the whole integration interval is divided into the minimum number of equal steps $\le h_{\max}$. In each primary current step, if $R_k > Tol$ the step is rejected, divided by two equal ones and the integrating procedure is repeated for each half recursively. In other case it is admitted.

The first strategy looks more flexible as the step length ratio is arbitrary, but not an integer power of 2. Really the result depends on the used quadrature and integrated function. One can only expect the first strategy should be more effective in the case of very smooth and homogeneous integrated function while the second one should be more effective in the case like (7), Fig. 1 or if the function contains several breaks of its values or derivatives. Of course, many modifications and combinations of these strategies are possible.

5. Optimal order of quadrature

The problem is which order of a quadrature should be used to integrate a given function in specified limits with a fixed accuracy. Consider the problem using the Gaussian quadrature and different integrated functions. Of course, the result can differ for another kind of quadrature, but not qualitatively. Let's estimate the local error by its main term, that is taking $f^{(k)}(0)$ instead of $f^{(k)}(\xi)$ in (5).

(9) $$R_k \sim c_k f^{(k)}(0).$$

If the whole integration interval $(-1,\ 1)$ is divided into elementary intervals of the length 2α each, their number is $1/\alpha$, the local error is

(10) $$R_k \sim c_k \alpha^{k+1} f^{(k)}(0)$$

as a consequence of

(11) $$f^{(k)}(x) = \alpha^{-k} f^{(k)}(\alpha x),$$

(12) $$\int_{-\alpha}^{\alpha} f(x)dx = \alpha \int_{-1}^{1} f(\alpha x)dx.$$

The global error in this case is

(13) $$G \le \left| c_k \alpha^k f^{(k)}(0) \right|.$$

Limiting the global error as Tol one gets

(14) $$\alpha \approx \left| \frac{Tol}{c_k f^{(k)}(0)} \right|^{1/k}$$

and the number of necessary function evaluations

(15) $$n\left|\frac{c_k f^{(k)}(0)}{Tol}\right|^{1/k},$$

where n is the number of function evaluation per step ($k/2$ for the Gaussian quadrature). Let's consider three types of functions: (i) something like $\cos$, $\exp$

(16) $$f^{(k)}(0) \sim 1,$$

(ii) something like $\ln$, $1/(1+x)$

(17) $$f^{(k)}(0) \sim k!,$$

and (iii) something irregular (its Taylor series diverges everywhere)

(18) $$f^{(k)}(0) \sim k!^2.$$

The error constant of the Gaussian quadrature is [8], 25.4.29

(19) $$c_k^G = \frac{2^{k+1}(k/2)!^4}{(k+1)k!^3}.$$

After the substitutions one obtains the necessary number of function evaluations in the three cases, that depends on the using order and the desired accuracy Fig. 2.

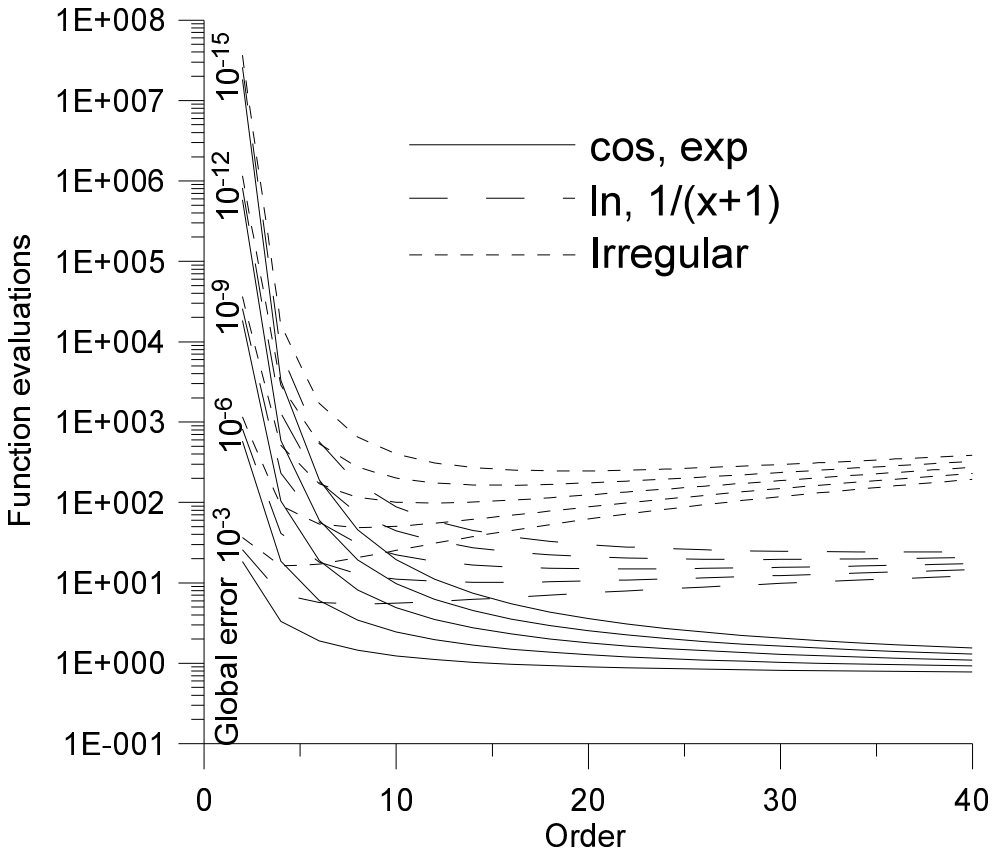

Fig. 2. Effectiveness of quadratures of various orders.

One can see, in the first case the higher the order, the greater the effectiveness. In other cases there are optimal order depending on the desired accuracy. It ranges between 4 and 40 in Fig. 2. Note that the effectiveness insufficiently depends on the order if the latter is ≥ 10. In real calculation, a higher order quadrature looks less flexible due to greater step length and nodes per step, so loss caused by a rejected step is larger. Thus, one can conclude, there is almost no sense to use quadratures of the order higher than 10 for any reasonable accuracy.

6. Design of integrating codes

Three codes were designed to test performance of the proposed quadratures. The first two using the Gauss-Kronrod quadrature and the Lobatto based one are very similar. In the first step and after admitted one the subsequent strategy is used. The tree-like strategy is applied if the current step (not first one) is rejected. Then the error estimation from the right border interval of the tree-like pattern is used to predict the next step length. The Lobatto-Kronrod quadrature and the Gauss based one were not tested as they have the same properties as the Lobatto based one except of lower order of the result quadrature. Additional advantage of the Lobatto quadrature (also

Lobatto-Kronrod and Lobatto based ones) containing an odd number of nodes in the case of the tree-like strategy is that they uses the central node in both halves of the divided interval, so that the necessary number of function evaluations is decreased by one for one step. This advantage is particularly dramatic in the case of the Newton-Cotes quadrature containing $2^i + 1$ (positive integer i) nodes. In the last case the necessary number of function evaluations is decreased twice. The drawbacks of these quadratures are that their error constants are significantly greater and all the reused nodes are to be stored.

The last code is based on the pure Lobatto quadrature and two ways of error estimation depending on the current step control strategy. An estimation using the nodes of the adjacent step as described in 3.3 and Appendix IV is ever available. The Richardson extrapolation can be used only in a tree-like procedure. As the quadrature claims two equal consequent steps to estimate the local error, it looks less flexible in comparison to the previous ones, so the consequent step control strategy should be more conservative, hence less effective, and the tree-like one is more preferable. So the tree-like procedure is applied each time when a step is to be rejected (including the first step). Unfortunately both the Richardson extrapolation and the adjacent step quadrature cannot estimate the local error separately for the right step and the left one. So only both steps can be admitted or rejected simultaneously. Another unpleasant thing is that the adjacent step quadrature can underestimate the local error if the integrated functions has a singularity within the current step.

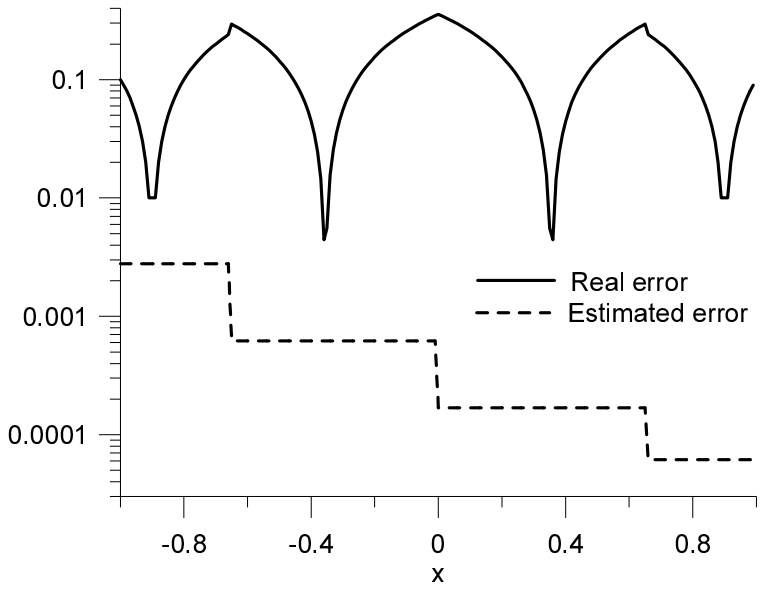

Fig. 3. The real error in calculation of $\int_{-1}^{1}\theta(x-\xi)d\xi$ using the Lobatto 8 quadrature and one estimated by the adjacent step quadrature described in 3.3 and Appendix IV.

The Richardson extrapolation experiences the same problem in this case. It uses a hypothesis $R_k \propto h^{k+1}$, but in this case $R_k \propto h$. A solution is to estimate the "local order". As the error estimation using the adjacent step quadrature is available both for the current step R_c and double-length one R_d, the "local order" can be estimated as

(20) $$\tilde{k} \sim \max\left(\min\left(\log_2\left(R_d / R_c\right)-1, k\right), 0\right).$$

Then $\tilde{k}$ is used instead of k in the Richardson extrapolation and the local error is estimated well.

Thus, the distinctive features of the last code are:

- The step length is changed only if it should be increased 1.8 or more times or decreased.
- The tree-like partitioning is used each time the local step is to be changed and in the very first step.
- The adjacent step quadrature is used by the consequent strategy ever and by the tree-like one to estimate the "local order".
- The Richardson extrapolation with the estimated "local order" is ever used by the tree-like strategy.

♦ When the consequent strategy is applied the Lobatto quadrature is used for the result. In the other case the Richardson extrapolation result is stored, so the order of the method is increased by two.

7. Testing problems

The following problems were used to check the effectiveness of each code (and quadrature!):

(21) $$\int_0^{100} \sin x dx \cong 0.13768112771231606589805 7.$$

(22) $$\int_0^{100} \sin^{100} x dx \cong 8.00118283137199704273636.$$

(23) $$\int_0^{10} \operatorname{sgn}(\sin x) dx \cong 2.56637061435917295385057.$$

(24) $$\int_{10^{-5}}^{1} \ln x dx \cong -0.99987487074535029771579 9.$$

(25) $$\int_{10^{-5}}^{1} \sin(1/x) dx \cong 0.50406706200686438117612 3.$$

(26) $$\int_{-1}^{1} x \sin(1/x) dx \cong 0.75706003424832261976346.$$

All the integrated functions have no singularities within the integration interval except of (24), that has breaks $x = l\pi$, integer l. The first one is homogenous enough, while the various orders derivatives of others vary dramatically on the integration interval. In (24) and (25) the obtained result depends on the direction of integration, but not so strongly. The best value is ever chosen.

8. Results of tests and discussion

The degrees of exactness available are: 8(4), 12(6), 14(8), and 18(10) in the code using the Gauss-Kronrod quadrature; 8(4), 12(6), 16(8), and 20(10) in the one using the Lobatto based quadrature (LobattoB in the figures below); 4(5), 8(9), 12(13), and 16(17) in the one using the pure Lobatto quadrature and the adjacent step

quadrature and the Richardson extrapolation for error estimation (DLobatto in the figures).

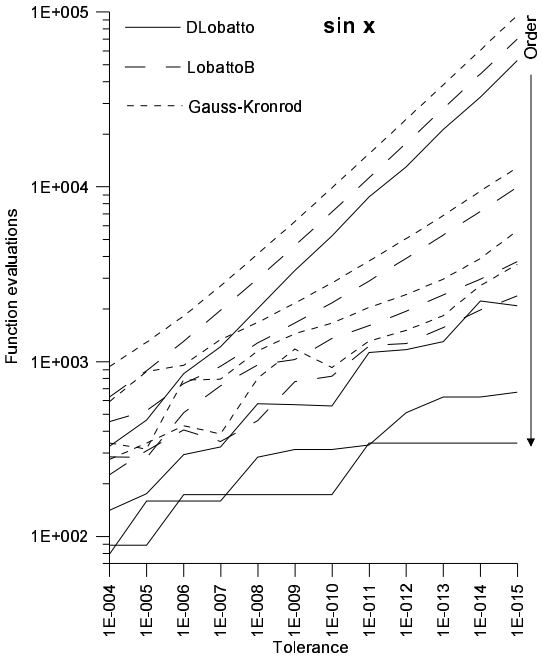

Fig. 4. Problem (21).

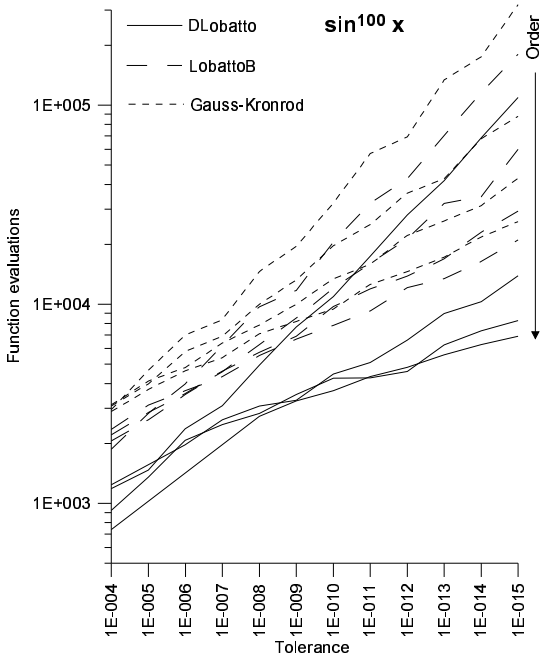

Fig. 5. Problem (22).

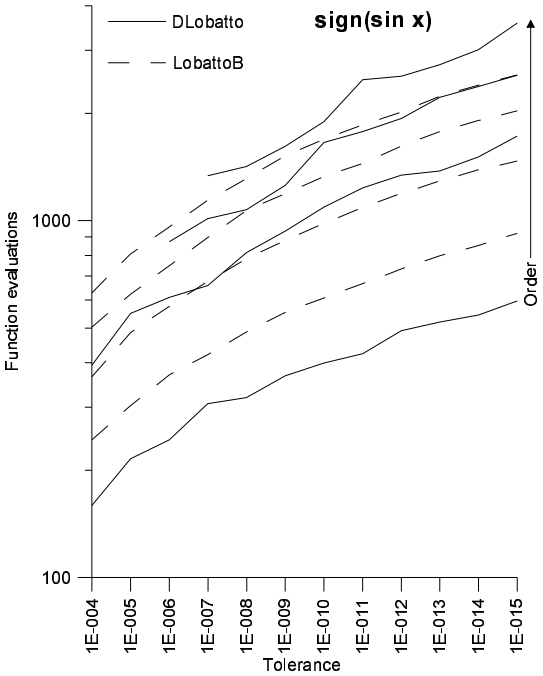

Fig. 6. Problem (23).

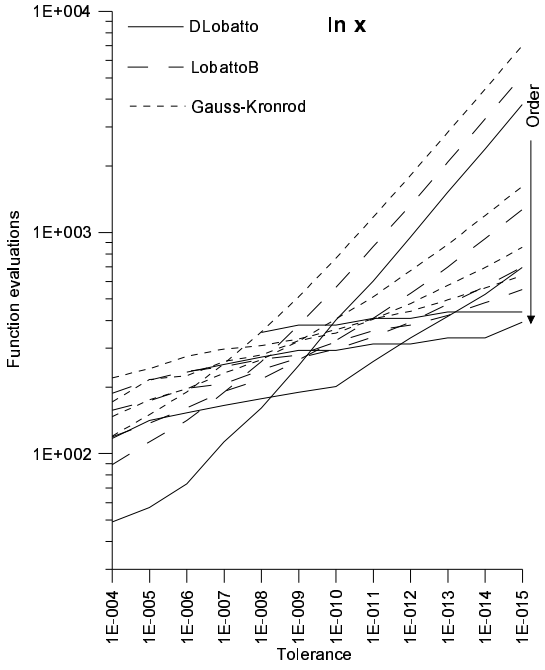

Fig. 7. Problem (24).

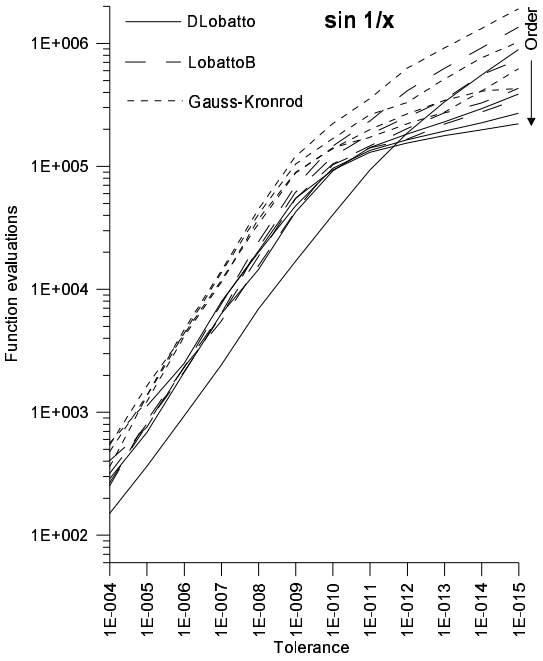

Fig. 8. Problem (25).

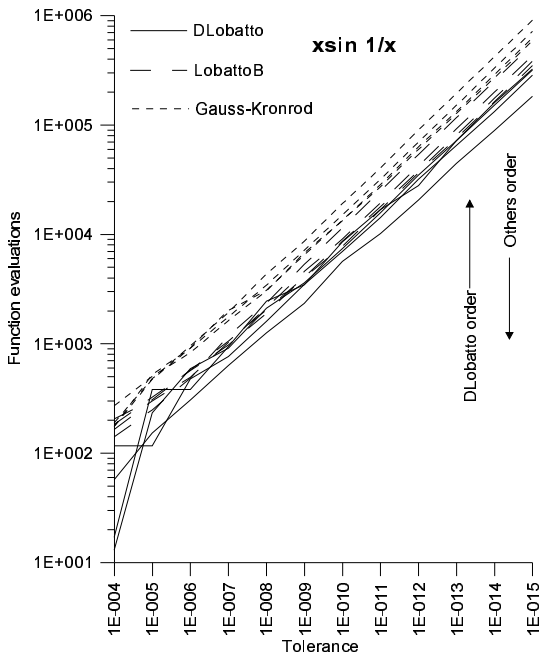

Fig. 9. Problem (26).

One can see in Fig. 4 - Fig. 9 that the DLobatto code is ever more effective than the LobattoB one, the latter is ever more effective than the Gauss-Kronrod one of the comparable order. Note that the tolerance value, i.e. the maximum local error specified, but not the real global error achieved is the argument of each horizontal axis. Of course, these two values are different. For example, in the problem (21) the real global error was ever 10^{-10} and less for the first two codes and $\sim Tol$ for the last one, but it indicates only not so good operation of the error estimator in the appropriate formulae. In other cases the global error was $\sim Tol$ or could vary within $2-3$ orders.

The Gauss-Kronrod code produced wrong results for the problem (23) most of all, so appropriate graphs were excluded from Fig. 6. The explanation of this fact is that the set of nodes of the Gauss-Kronrod quadrature does not cover the whole integrating interval. The situation is possible when the break of the integrated function is placed between the end of the interval and the extreme node, so the local error can be arbitrarily large but the difference between the two quadratures in the pair is small enough (0 in the mentioned problem). This phenomenon is impossible in the last two codes as the nodes of their quadratures totally cover the integrated interval.

The DLobatto code experienced some troubles in solving the problems (22) - (24) if a high order and comparably large $Tol \sim 10^{-4} - 10^{-7}$ were chosen. The

explanation is that the error estimation using the two-step Lobatto based quadrature is not so reliable in the case of not so smooth function. It is due to extremely small coefficients in the difference of the mentioned quadrature and the base Lobatto one. This trouble was never observed in the steps with the Richardson extrapolation. An obvious conclusion is to use higher order methods only with small enough Tol. Note that a lower order formula is ever effective for the problem (23).

9. Conclusions

Four new quadratures with local error estimation were proposed and two of them were numerically tested together with the combination step control algorithm. They provides higher effectiveness and robustness than the well known Gauss-Kronrod one. The Lobatto-Kronrod quadrature and the Lobatto based one have very similar properties, but the latter has higher order of the result and less number of function evaluations per step, all other things being equal. The two-step Lobatto based quadrature looks more effective, but no so robust as the previous two. The proposed choices are the Lobatto based quadrature for the most robustness and two-step Lobatto-based one for the most effectiveness. In the last case do not use a high order with a moderate tolerance $10^{-4} - 10^{-7}$. Probably the most universal choice is the two-step Lobatto based quadrature $8(9)$.

References

[1] L.F.Richardson. The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam. Phil. Trans., **A210** (1910), 307-357.

[2] L.F.Richardson. The deferred approach to the limit. Phil. Trans., **A226** (1927), 299-349.

[3] A.S.Kronrod. Integration with control of accuracy. Soviet Physics Dokl., **9** (1964), #1, 17-19. Original paper (in Russian):
А.С.Кронрод. об интегрировании с контролем точности. Доклады АН СССР, **154** (1964), №2, 283-286.

[4] A.S.Kronrod. Nodes and weights of quadrature formulas. Sixteen-place tables. Authorized translation from the Russian, Consultants Bureau, New York, 1965. Original book:

А.С.Кронрод. Узлы и веса квадратурных формул. Шестнадцатиразрядные таблицы. М., Наука, 1964.

[5] S.Ehrich. Error bounds for Gauss-Kronrod quadrature formulae. Math of Comput., **62** (1994), #205, 295-304.

[6] S.Ehrich. Asymptotic properties of Stieltjes polynomials and Gauss-Kronrod quadrature formulae. J.Approx. Theory, **82** (1995), #2, 287-303.

[7] D.P.Lauric. Anti-Gaussian quadrature formulas. Math. Comput., **65** (1996), #214, 739-747.

[8] Handbook of Mathematical Functions. National Bureau of Standards, Applied Mathematics Series 55, 1964.

Appendix I. Coefficients of Gauss based quadratures

The Gauss based quadrature contains all the n nodes of Gaussian one except of the border two, and the nodes ± 1. Values in the headers means the degrees of exactness. The nodes and the coefficients of the Gaussian quadratures see in [8], Table 25.4.

2(4) is the Lobatto 2 or Newton-Cotes 2 quadrature or the rule of trapezium.

4(6) is the Newton-Cotes 4 or Simpson or Lobatto 4 quadrature.

6(10) $c_6 \cong 5.13 \cdot 10^{-5}$

$$\frac{32}{45} - \frac{8\sqrt{70}}{225}, \quad \frac{7}{9} - \frac{7\sqrt{70}}{450}, \quad \frac{\sqrt{70}}{30} - \frac{2}{15}.$$

8(14) $c_8 \cong 1.48 \cdot 10^{-7}$

$0.49122808604223767619672 3,\ 0.30520527143593210845070 9,$
$0.36904282357710873796790 6,\ 0.08013786196584031548302 3.$

10(18) $c_{10} \cong 2.93 \cdot 10^{-10}$

$0.28922778533166782139378 2,\ 0.35451024377484733689404 5,$
$0.21448170649241905373888 8,\ 0.23593070220001898906945 8,$
$0.05046345486688070960071 7.$

Appendix II. Nodes and coefficients of Lobatto-Kronrod quadratures

The Lobatto-Kronrod quadrature contains all the $n+1$ nodes of the Lobatto one and n additional nodes. It is symmetric. Values in the headers means the degrees of exactness.

4(2) is the Simpson quadrature.	
8(4) is the Lobatto 8 quadrature	
10(6): $c_{10} \cong 1.70 \cdot 10^{-10}$	
Nodes	Coefficients
0	$16/35$
$\sqrt{5}/5$	$125/294$
$\sqrt{6}/3$	$72/245$
1	$11/210$
14(8): $c_{14} \cong 6.16 \cdot 10^{-16}$	
0	$47104/137025$
$\sqrt{\frac{5}{11}-\frac{6\sqrt{65}}{143}}$	$\frac{46299523}{1802152800}+\frac{2474329\sqrt{65}}{257450400}$
$\sqrt{21}/7$	$16807/59184$
$\sqrt{\frac{5}{11}+\frac{6\sqrt{65}}{143}}$	$\frac{46299523}{1802152800}-\frac{2474329\sqrt{65}}{257450400}$
1	$139/4536$
16(10): $c_{16} \cong 2.03 \cdot 10^{-19}$	
0	$43264/148995$
$\sqrt{\frac{1}{3}-\frac{2\sqrt{7}}{21}}$	$\frac{19203149}{81648765}+\frac{2487779\sqrt{7}}{163297530}$
$\sqrt{\frac{15}{26}-\frac{\sqrt{53}}{26}}$	$\frac{75969518144}{417796730505}+\frac{8720929984\sqrt{53}}{1054439367465}$
$\sqrt{\frac{1}{3}+\frac{2\sqrt{7}}{21}}$	$\frac{19203149}{81648765}-\frac{2487779\sqrt{7}}{163297530}$
$\sqrt{\frac{15}{26}+\frac{\sqrt{53}}{26}}$	$\frac{75969518144}{417796730505}-\frac{8720929984\sqrt{53}}{1054439367465}$
1	$1223/58905$

Appendix III. Coefficients of Lobatto based embedded quadratures

The Lobatto based quadrature contains all the nodes of the Lobatto one except of -1 and 1. Values in the headers means the degrees of exactness. The nodes and the coefficients of the Lobatto quadratures see in [8], Table 25.6.

2(4) is the Gaussian 2 quadrature or the middle point rule.

4(8): $c_4 = -1/210$.

$4/9,\ 7/9$.

6(12): $c_6 \cong -3.21\cdot 10^{-5}$.

$0.64,\ 0.28836022205056777486 4277,\ 0.39163977794943222513 5723$.

8(16): $c_8 \cong -1.23\cdot 10^{-7}$.

$0.26993197278911564625 8503,\ 0.44452547419666367952 1031,$

$0.18721130931406719497 772,\ 0.23329723009471130237 1994$.

10(20): $c_{10} \cong -3.05\cdot 10^{-10}$.

$0.37409926933736457545 9813,\ 0.21465735460621977202 6581,$

$0.31520438120128297356 7436,\ 0.12883388294903539270 4926,$

$0.15425474657477957397 1146$

Appendix IV. Coefficients of two-step Lobatto based quadratures

The two-step Lobatto based quadrature contains all the nodes of the Lobatto ones in the steps $[-3,\ -1]$ and $[-1,\ 1]$. Its error constant has the same absolute value, but the opposite sign as one of the Lobatto quadrature. The nodes and the coefficients of the Lobatto quadratures see in [8], Table 25.6.

5(4):

$-1/90,\ 2/45,\ 4/15,\ 62/45,\ 29/90$.

9(8):

$$-\frac{1}{8100},\ \frac{49}{13500}-\frac{7\cdot\sqrt{21}}{10125},\ -\frac{16}{10125},\ \frac{49}{13500}+\frac{7\cdot\sqrt{21}}{10125},\ \frac{4}{45},\ \frac{7399}{13500}+\frac{7\cdot\sqrt{21}}{10125},$$

$$\frac{7184}{10125},\ \frac{7399}{13500}-\frac{7\cdot\sqrt{21}}{10125},\ \frac{809}{8100}.$$

13(12):

$-2.0282412308990\cdot 10$, $6.9261275747576\cdot 10$, $-1.965977375615296\cdot 10^{-5}$,

$7.5684337185547\cdot 10^{-5}$, -0.0003923869878066355, $0.0021629663695352 1352$,

$0.0439560439560439560 4$, $0.2789890137311011615 3$, $0.4313529942220559879 4$,

$0.4876947319562331668 8$, $0.4317257214361064704 5$, $0.2768329734891407056 2$,

$0.0476170193778167200 1$.

17(16):

$-3.929175511241\cdot10^{-}$, $1.2627332094564\cdot10^{-}$, $-3.1129527828904\cdot10^{-7}$,
$9.740508127787\cdot10^{-7}$, $-4.07955787233817\cdot10^{-6}$, $2.262419161516132\cdot10^{-}$,
-0.00015235387263192787 , 0.00095005296584106966 ,
0.02614379084967316514 , 0.16644541452664659413 , 0.27438635862752980773 ,
0.34645113516466150632 , 0.37151519481854342496 , 0.34642948502385912380 ,
0.27453840120488344627 , 0.16549548783412647067 , 0.02777773848602266537 .