(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Simulation Modelling Practice and Theory 19 (2011) 1860–1872

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Numerical integration of sparsely sampled data

Natalia Petrovskaya [a,*], Ezio Venturino [b]

[a] School of Mathematics, University of Birmingham, Birmingham B15 2TT, United Kingdom
[b] Dipartimento di Matematica "Giuseppe Peano", Universita' di Torino, via Carlo Alberto 10, 10123 Torino, Italy

ARTICLE INFO

Article history:
Received 28 February 2011
Received in revised form 4 May 2011
Accepted 7 May 2011

Keywords:
Numerical integration
Sampled data
Coarse mesh

ABSTRACT

In experimental work as well as in computational applications for which limited computational resources are available for the numerical calculations a coarse mesh problem frequently appears. In particular, we consider here the problem of numerical integration when the integrand is available only at nodes of a coarse uniform computational grid. Our research is motivated by the coarse mesh problem arising in ecological applications such as pest insect monitoring and control. In our study we formulate a criterion for assessing mesh coarseness and demonstrate that the definition of a coarse mesh depends on the integrand function. We then discuss the accuracy of computations on coarse meshes to conclude that the conventional methods used to improve accuracy on fine meshes cannot be applied to coarse meshes. Our discussion is illustrated by numerical examples.

1. Introduction

In the general problem of numerical integration the approach chosen for the evaluation of the integral $I = \int_a^b f(x)\,dx$ is usually based on the definition of the integrand $f(x)$. In many cases the function $f(x)$ is an analytically defined function and the value $f(x^*)$ is available at any point $x^* \in [a, b]$. For a large class of problems, however, a method of numerical integration should deal with computation of integrals where the integrand is only available at discrete points x_n, $n = 0, 1, \ldots, N$. It is often the case that the values of the integrand $f_n = f(x_n)$ come from experimental data, as numerical integration of sampled data is required in many practical problems such as image reconstruction [20], optical interferometry [30], the growth of microorganisms [10] and various ecological applications [21]. The data $\{f_n\}$ defined over the set $\{x_n\}$ may also be generated as a result of computer simulation of a physical problem. Below we always refer to the set $\{x_n\}$ as a computational grid, no matter how the discrete data $\{f_n\}$ are obtained.

In spite of the many accurate and efficient methods for numerical integration of sampled data being available and documented in the literature [11,12,26], in the past decade the increasing complexity of practical problems resulted in the need to revise existing integration algorithms as new problems have emerged. One such problem is how to minimize the number of field measurements required to compute the integral I with an acceptable accuracy for the given practical problem. It is obvious that in practical applications the accuracy of integration depends on the number of detectors used to collect experimental data $\{f_n\}$, so that using a small number of detectors may provide inaccurate results. On the other hand, installation of additional detectors can make the evaluation of the integral I much more expensive and therefore beyond the budget allocated for the experimental measurements. Also, in some practical problems it is impossible to install detectors close to each other, as they begin to interfere and therefore contaminate the data measurements.

From the mathematical viewpoint installing a small number of detectors in a domain where sampled data are collected means that the discrete function $\{f_n\}$ is defined on a very coarse uniform grid. The generation of a uniform mesh is an important restriction that must be taken into account when the integral is evaluated. Nonuniform (adaptive) meshes cannot be

* Corresponding author. Tel.: +44 121 4146591; fax: +44 121 4143389.
E-mail addresses: n.b.petrovskaya@bham.ac.uk (N. Petrovskaya), ezio.venturino@unito.it (E. Venturino).

1569-190X/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2011.05.003

considered in the problem because of the practical requirement that the detectors used in field measurements should be equidistant. Moreover, it is often the case that a researcher cannot repeat measurements with a larger number of detectors in order to simulate uniform grid refinement.

The above restrictions make the question of the accuracy of numerical integration on a coarse mesh a difficult and challenging issue. Researchers conducting experimental work often apply a heuristic approach where some a priori information is used to define a coarse mesh in the problem. For instance, there is widespread reasoning in ecological applications [21,24] that the grid should resolve all function 'humps' (that is, regions where local extrema are present) where the number of humps along with the width of each hump can be evaluated beforehand from the knowledge of the diffusion coefficient in the problem. That gives a mesh cell size evaluation like 'N detectors per hump' for practical computations, where N is an ad hoc parameter. Obviously, this heuristic approach should be checked against a more rigorous definition of mesh coarseness that involves an integral error estimate.

Based on the above, the goal of our study is to investigate the problem of numerical integration of sparse data. In particular, our research has been motivated by a coarse mesh problem arising in ecological applications. The approach we suggest in the paper is based on the following argument. We consider a method of numerical integration (called the GLI method in the paper) whose accuracy depends on the degree of a polynomial used in the method formulation. Error estimates available in the problem predict that exploiting higher order polynomials should result in a smaller integration error in comparison with the error of Simpson's quadrature rule widely used on uniform meshes. Hence one may expect that the method will provide a more accurate integral evaluation on any uniform mesh. The reliability criterion then consists in comparing the error of the Simpson rule with the error of the GLI quadrature: if the former is smaller than the latter we conclude that the integrand function is not well resolved on the given mesh and a finer mesh is required for computation of the integral. It will be shown in the paper that a threshold N^* for the number of grid nodes exists, assessing the reliability of the numerical integration. Our approach will finally be illustrated by ecologically meaningful examples.

2. Numerical integration in ecological problems

In this section we briefly explain how the coarse mesh problem arises in ecological applications. Obtaining accurate information about the population size for a given pest species is crucial in many ecological problems. For instance, optimization of the use of pesticides remains one of the most important ecological tasks, and a decision about application of pesticides is usually made based on the information about the population size, where a certain threshold number should indicate the pest abundance in the region of interest (e.g., over an agricultural field) [27]. In order to get an estimate of the population size, i.e., the total number of the pest insects in the field, the population density should be integrated over the whole monitored area by a chosen quadrature method. Thus numerical integration of ecological data is an essential component of the ecological monitoring program.

Trapping is a common method to collect field data when pest insects are considered. The insect traps are installed in an agricultural field, where they are exposed for a certain time, after which the caught insects are counted to give the pest population density at the position of the traps [7]. For financial and agricultural reasons, the number of traps installed at nodes of a rectangular grid over a typical agricultural field is usually taken between 5 and 10 in each direction [1,4,17]. Hence, the problem of pest monitoring requires numerical integration of a discrete function obtained on a very coarse mesh. A question arising in ecological applications concerns the smallest number of traps that will provide the user with a reliable estimate of the integral.

The conventional problem of numerical integration has a long history and many efficient methods have been developed to compute the integral of a given function with a required accuracy (e.g., [9]). However, contrary to the standard problem where a computational grid can be refined to ensure the required accuracy of the integration, the grid of traps in a routine monitoring procedure cannot be refined. Firstly, installation of a large number of traps causes a strong disturbance to the agricultural field and may considerably damage the crop. Another reason is that trapping is a relatively expensive procedure, while the labor and financial resources available in the ecological monitoring program are usually restricted.

Numerical integration on coarse grids is a challenging mathematical problem, as most of the results of the numerical integration theory do not apply to the coarse grid integration. It will be shown in our paper that, while the conventional approach to the accuracy of numerical integration relies upon the asymptotic error estimates, the conclusions about the accuracy of integration on coarse meshes cannot be derived from the knowledge of error estimates. A different technique is required to obtain sufficiently accurate estimates of pest abundance keeping the resources involved in field data collection at the minimum possible value. In the next section we introduce a method of numerical integration required for rigorous formulation of a coarse grid problem. It shows that existing methods of numerical integration should be revisited when a coarse grid problem is considered.

3. Methods of numerical integration on coarse meshes

3.1. Motivation for using the GLI method

We focus our effort on the issue of reliability of numerical integration for one-dimensional coarse meshes. Consider a set of points x_n, $n = 0, 1, \ldots, N$ in the domain $[a, b]$. The points x_n are originally thought of as the coordinates of detectors installed

in a physical domain of interest for data collection. We require that $x_0 = a$, $x_{n+1} = x_n + h$, $n = 0, \ldots, N-1$ where the grid step size h is defined as $h = (b-a)/N$, and refer to the union of subintervals $e_n = [x_n, x_{n+1}]$ as a uniform computational grid $\mathcal{G}_{[a,b]}$ with nodes x_n, $n = 0, \ldots, N$. We also assume that the data $f_n = f(x_n)$ are known at each point x_n and only at these points.

The general problem of numerical integration is that the integral $I = \int_a^b f(x)\,dx$ should be approximated by the sum $I_a = \sum_{l=1}^{L} f(x_l)C_l$ to meet the condition $|I - I_a| < \epsilon$, where ϵ is a given tolerance. In many problems of numerical integration the value of the function $f(x)$ can be computed at any arbitrary point x. It is therefore possible to choose the sample points x_l and weights of integration C_l in an optimal way so as to achieve the desired accuracy. Moreover, efficient algorithms have been developed to offer adaptive approaches to numerical integration, where the number of points over a computational grid is gradually increased in those subregions where the accuracy of integration is not good enough [13,14,16].

Here, however, we have to deal with an essentially different type of problem where the integrand $f(x)$ is only available at points x_n of a uniform grid $\mathcal{G}_{[a,b]}$, the number N of grid cells is small (a coarse mesh) and the mesh cannot be adaptively refined. Thus one part of the problem is to find a method that will allow us to integrate the function $f(x)$ with the maximum achievable accuracy for the sparsely sampled data available for the problem. The Newton–Cotes integration rules are a class of integration formulas widely used on uniform meshes [3,9,12]. The $i+1$-point Newton–Cotes rule on a uniform grid $\mathcal{G}_{[a,b]}$ is given by (e.g., see [9])

$$\int_a^b f(x)\,dx \approx h\sum_{j=0}^{i} B_{ij} f(x_j),$$

where

$$B_{ij} = \frac{(-1)^{i-j}}{j!(i-j)!}\int_0^i t(t-1)\cdots(t-j+1)(t-j-1)\cdots(t-i)\,dt.$$

Since the weights of the higher order Newton–Cotes formulas are not always positive, increasing the number i of nodes in the Newton–Cotes formulas does not necessarily lead to a more accurate integral value, in view of possible cancellations. The example considered in Ref. [9] shows that increasing the number i from $i = 2$ up to $i = 21$ results in a larger integration error. Moreover, it is often difficult to apply a composite Newton–Cotes rule with $i > 2$ on a uniform mesh with an arbitrary number of mesh subintervals, as the total number N of mesh cells is required to be a multiple of i. That is why in many experimental applications the integral evaluation is restricted by the use of the composite trapezoidal rule or the composite Simpson rule which represent the first two rules in the Newton–Cotes family. Hence the first task of our study is to design a method of numerical integration that will mainly be aimed for integration on a coarse mesh and will possess the following properties:

(i) The method should be more accurate than the composite Simpson rule.
(ii) The method should not have any restrictions with respect to the number of grid nodes, unlike higher order Newton–Cotes formulas, whose implementation is limited as discussed above. In other words, the method can be applied on a mesh with a relatively small but arbitrary number of grid nodes without losing accuracy.
(iii) The method's applicability should not be restricted by a uniform mesh. In particular, one should be able to apply the method if there are small deviations of grid nodes from their position on a uniform grid, a fact that often happens in experimental measurements.

Keeping in mind the above requirements, one natural solution to the problem is to consider polynomials used to approximate the integrand function separately in each grid subinterval. In the following subsection we discuss how to combine the local polynomial interpolation with the Gauss–Legendre formula in order to increase the accuracy of integration. Let us note here that the method we discuss below is obviously not the only option that a user may implement for integration on coarse meshes (cf. [2,19]). Our choice of the method has mostly been motivated by its flexibility, i.e., increasing a polynomial degree in the integrand approximation should allow one to obtain a more accurate value of the integral.

3.2. The Gauss–Legendre rule with interpolation

Consider the K-point Gauss–Legendre rule given by

$$\int_a^b f(x)\,dx \approx I_a = \frac{b-a}{2}\sum_{k=1}^{K} w_k f\left(\frac{a+b}{2} + \tilde{\xi}_k\frac{b-a}{2}\right),$$

where the points $\tilde{\xi}_k$ are the roots of the Legendre polynomials computed on the interval $[-1, 1]$ and the weights w_k are defined by the set $\{\tilde{\xi}_k\}$. We can also apply a composite Gauss–Legendre rule to the problem as follows

$$\int_a^b f(x)\,dx \approx \sum_{n=0}^{N-1} I_a^n, \quad I_a^n = \frac{x_{n+1}-x_n}{2}\sum_{k=1}^{K} w_k f\left(\frac{x_n + x_{n+1}}{2} + \tilde{\xi}_k\frac{x_{n+1}-x_n}{2}\right),$$

where I_a^n has to be computed on each subinterval $e_n = [x_n, x_{n+1}]$. It is well-known (e.g., see [9]) that the general K-point Gauss–Legendre rule is exact for polynomial functions of degree $\leqslant 2K-1$. In other words, if we have a sufficiently smooth integrand

function on a computational grid with a fixed number of grid nodes, we may expect that the Gauss–Legendre rule on that grid would be more accurate than the K-point Newton–Cotes rule. The main obstacle in implementation of the Gauss–Legendre rule in our problem is that the algorithm requires information about the integrand function at the abscissae of the Gauss–Legendre formula. The data at the roots of the Legendre polynomials are not available to us by the problem statement, as we have to use a uniform grid only. However, we can interpolate the data sampled on a uniform mesh and use the interpolated values at the Gaussian nodes for the numerical integration. The crucial requirement for such approach is that the interpolation error arising when we transfer the data from a uniform mesh to the abscissae of the Gauss–Legendre quadrature should not exceed the integration error of the numerical integration method based on the same uniform grid. We explain this requirement in more detail below.

Consider the integral $I = \int_{X_1}^{X_2} f(x)\,dx$ in the domain $[X_1, X_2]$, where $[X_1, X_2] \subset [a, b]$. The standard approach in numerical integration is to expand the integrand $f(x)$ as

$$f(x) = \sum_{k=1}^{K} f(x_k)\phi_k(x) + r(x),$$

where $\phi_k(x)$ are polynomial basis functions, $f(x_k)$ are function values at interpolation nodes, and $r(x)$ is the remainder term. Substituting the above expansion into the integral we arrive at

$$I = \int_{X_1}^{X_2} \left(\sum_{k=1}^{K} f(x_k)\phi_k(x) + r(x) \right) dx = \sum_{k=1}^{K} f(x_k) \int_{X_1}^{X_2} \phi_k(x)\,dx + \int_{X_1}^{X_2} r(x)\,dx.$$

Thus the integral is

$$I = \sum_{k=1}^{K} f(x_k)C_k + R,$$

where weights are $C_k = \int_{X_1}^{X_2} \phi_k(x)\,dx$, and the integral remainder is given by $R = \int_{X_1}^{X_2} r(x)\,dx$.

Let us now assume that we do not know the exact values $f(x_k)$, $k = 1, \ldots, K$. Instead, the function $f(x)$ is defined with the error ϵ_k at each point x_k,

$$f(x) = \sum_{k=1}^{K} (f(x_k) + \epsilon_k)\phi_k(x) + r(x).$$

We then have

$$\begin{aligned} I &= \int_{X_1}^{X_2} \left(\sum_{k=1}^{K} (f(x_k) + \epsilon_k)\phi_k(x) + r(x) \right) dx = \sum_{k=1}^{K} (f(x_k) + \epsilon_k) \int_{X_1}^{X_2} \phi_k(x)\,dx + \int_{X_1}^{X_2} r(x)\,dx \\ &= \sum_{k=1}^{K} f(x_k) \int_{X_1}^{X_2} \phi_k(x)\,dx + \sum_{k=1}^{K} \epsilon_k \int_{X_1}^{X_2} \phi_k(x)\,dx + \int_{X_1}^{X_2} r(x)\,dx. \end{aligned}$$

Hence, the integral is

$$I = \sum_{k=1}^{K} f(x_k)C_k + R^*, \tag{1}$$

where the new remainder R^* is given by

$$R^* = R_\epsilon + R, \quad R_\epsilon = \sum_{k=1}^{K} \epsilon_k \int_{X_1}^{X_2} \phi_k(x)\,dx = \sum_{k=1}^{K} \epsilon_k C_k. \tag{2}$$

Let a uniform mesh $\mathcal{G}_{[X_1,X_2]}$ of $p+1$ nodes be generated over $[X_1,X_2]$ as a subset of the original grid $\mathcal{G}_{[a,b]}$. Namely, we require that $\hat{x}_0 = X_1 \equiv x_i$, $\hat{x}_{q+1} = \hat{x}_q + h, \hat{x}_p = X_2 \equiv x_{i+p}$, where $\hat{x}_q \in \mathcal{G}_{[X_1,X_2]}, q = 0, \ldots, p,\ x_s \in \mathcal{G}_{[a,b]}, s = i, \ldots, i+p$ and $h = (b-a)/N$. Now let us interpolate the function $f(x)$ from the uniform mesh $\mathcal{G}_{[X_1,X_2]}$ at points ξ_k, $k = 1, \ldots, K$ where the location of points $\{\xi_k\}$ on grid $\mathcal{G}_{[a,b]}$ will be discussed later in the text. The function $f(x)$ is then defined at each point ξ_k with the interpolation error ϵ_k and, according to (2), the remainder R_ϵ will be controlled by the error ϵ_k.

The interpolation error ϵ_k depends on the method chosen to approximate the function $f(x)$. We use Lagrange interpolation in order to compute the function values at points $\{\xi_k\}$. The Lagrange interpolation polynomial of degree p is defined on the interval $[X_1, X_2]$ as

$$L_p(x) = \sum_{q=0}^{p} f(\hat{x}_q) \prod_{l \neq q} \frac{x - \hat{x}_l}{\hat{x}_q - \hat{x}_l}, \tag{3}$$

where $f(\hat{x}_q),\ q = 0, \ldots, p$ are the values of the function $f(x)$ at selected points $\hat{x}_q \in [X_1, X_2]$. For our problem, the points $\hat{x}_q,\ q = 0, \ldots, p$ are nodes of the uniform grid $\mathcal{G}_{[X_1,X_2]}$. The error of the Lagrange interpolation is given by

$$|f(x) - L_p(x)| \leqslant \frac{(X_2 - X_1)^{p+1}}{(p+1)!} \sup_{x \in [X_1, X_2]} \left|f^{(p+1)}(x)\right|, \qquad \forall x \in [X_1, X_2]. \tag{4}$$

Furthermore, let us assume that $f(x)$ and the derivatives of $f(x)$ up to the order $p+1$ are continuous and bounded on any subinterval $\left[\hat{x}_q, \hat{x}_{q+s}\right]$, $1 \leqslant s \leqslant p$,

$$\left|f^{(p+1)}(x)\right| \leqslant D_{s+1} \quad \text{for} \quad \hat{x}_q < x < \hat{x}_{q+s},$$

where D_{s+1}, $s = 1, \ldots, p$ are constants. Then the error of the Lagrange interpolation is

$$\left|R_p(x)\right| \leqslant C_p h^{p+1}.$$

Thus the remainder R_ϵ in (2) induced by the Lagrange interpolation with a fixed polynomial degree p is

$$R_\epsilon = \sum_{k=1}^{K} \epsilon_k \int_{X_1}^{X_2} \phi_k(x)\, dx \leqslant C_\epsilon h^{p+1}, \tag{5}$$

where the constant reads $C_\epsilon = C_p \sum_{k=1}^{K} C_k$.

Let now ξ_k, $k = 1, \ldots, K$, be the abscissae of the Gauss–Legendre formula taken in a subinterval $e_n = [x_n, x_{n+1}]$ of the grid $\mathcal{G}_{[a,b]}$. We interpolate the values of the function $f(x)$ from a uniform mesh $\mathcal{G}_{[X_1,X_2]}$ to points ξ_k as $f(\xi_k) = L_p^n(\xi_k)$, where the notation $L_p^n(x)$ means that a Lagrange polynomial of degree p is designed for interpolation of the function $f(x)$ on the points $\{\xi_k\}$, $k = 1, \ldots, K$, that belong to the subinterval e_n. The choice of nodes x_i through x_{i+p} (in other words, the choice of the subinterval $[X_1, X_2]$) used for the local Lagrange interpolation will then depend on the definition of the subinterval e_n, and we will refer further in the paper to the grid $\mathcal{G}_{[X_1,X_2]}$ as the local interpolation stencil for e_n.

Once the function values $f(\xi_k)$ have been defined at $e_n = [x_n, x_{n+1}]$, we can apply the Gauss–Legendre integration rule on the subinterval e_n and get the estimate $I_a^n \approx \int_{x_n}^{x_{n+1}} f(x)\, dx$. We then compute the integral over the domain $[a, b]$ as

$$I_a = \sum_{e_n} I_a^n. \tag{6}$$

The evaluate I_a^n requires a local interpolation stencil $\mathcal{G}_{[X_1,X_2]}$ that can be generated independently for each subinterval $e_n = [x_n, x_{n+1}]$, $n = 0, \ldots, N-1$. Thus the compound interpolation rule (6) can be applied on a uniform grid $\mathcal{G}_{[a,b]}$ with any number N of grid subintervals, where N should not be a multiple of $p+1$.

Since the remainder of the K-point Gauss–Legendre rule at each subinterval e_n is given by (e.g., see [9])

$$R = \frac{(K!)^4 f^{(2K)}(\eta)}{(2K+1)[(2K!)]^3} h^{2K+1}, \quad x_n < \eta < x_{n+1},$$

the error (5) becomes dominant in the evaluation (2) for $K > p/2$. The error (5) should then be compared with the error R_u of a numerical integration method on a uniform mesh. If we use the compound Simpson rule, then the error R_u is given by

$$|R_u(x)| \leqslant C_S h^4, \tag{7}$$

where C_S is a constant [9]. It can be readily seen from (5) and (7) that the polynomial degree in (3) should be chosen to make the error (5) smaller than the one in Simpson's method, thus we require $p \geqslant 4$. This should result in more accurate numerical integration in comparison with the Simpson rule used on the same mesh.

The algorithm for the GLI method is explained in detail in Appendix A. Let us emphasize it again that the method described above is, of course, not the only option for numerical integration when local polynomial approximation of an integrand function is considered. Analytical integration of locally defined Lagrange polynomials (3) would bring us the same results in terms of method's accuracy and flexibility. However, integrating a family of higher order polynomials in closed form is a laborious task, while in our approach we readily exploit a standard Lagrange interpolation and a Gaussian quadrature computer program available at [5].

4. Numerical results: A coarse mesh problem

In this section we first consider a number of numerical test cases that should help us to formulate a coarse mesh problem. In the test cases below the integrals are computed on a single uniform mesh defined by the requirement that the unit interval [0,1] should contain $N = 8$ subintervals. The number of subintervals for an arbitrary closed interval $[a, b]$ is then evaluated by rounding off the length of $[a, b]$ to the integer number. For example, if the integral should be computed on the interval $[0, \pi]$, then a computational mesh will contain $N = 24$ subintervals.

Table 1 displays conventional test cases often used to validate a numerical integration approach. The test cases in the table have been taken from the papers [18,28]. The notation $a{:}b$ gives the lower and the upper limit of integration, respectively. The function $f(x)$ is the integrand and I is the exact value of the integral. It is worth mentioning here that for the discussion in our paper we found it sufficient to carry out testing of smooth functions only (those presented in group A, group C and group

Table 1
The numerical integration test cases.

Testcase	a:b	$f(x)$	I
1	0:1	x^3	1/4
2	0:1	x^6	1/7
3	0:1	x^{10}	1/11
4	0:1	$\sqrt{x^7}$	2/9
5	0:π	$\sin(x)$	2
6	1:10	$1/x$	$\ln(10)$
7	0:1	e^x	$e-1$
8	0:1	$x/(e^x-1)$	7.7750463411e−01
9	−4:4	$1/(1+x^2)$	$2\tan^{-1}(4)$
10	−1:1	$1/(x^4+x^2+0.9)$	1.5822329637e+00
11	0:$\pi/2$	$1/(1+\sin^2 x)$	$\pi\sqrt{2}/4$
12	0:5	$\frac{1}{120}(x-1)(x-2)\ldots(x-5)$	-47.5/144.0
13	−1:1	$\frac{23}{25}\cosh(x)-\cos(x)$	$\frac{46}{25}\sinh(1)-2\sin(1)$
14	0:1	$\frac{1}{(x+0.01)^5}$	$\frac{(0.01)^{-4}-(1.01)^{-4}}{4}$
15	0:1	$1/\sqrt{x+0.0001}$	$2.0\sqrt{1.0001}-\frac{1}{50}$
16	0:1	$1/(x+0.0001)$	$\ln(10001.0)$
17	0:1	$1/((230x-30)^2+1)$	$\frac{1}{230}(\tan^{-1}(200)+\tan^{-1}(30))$
18	0:1	$1/(x+0.01)$	$\ln(101)$
19	0:10	$50/(\pi(1+2500x^2))$	$\tan^{-1}(500)/\pi$
20	0:1	$2/(2+\sin(10\pi x))$	1.1547006690
21	0.1:1	$(\sin(100\pi x)/(\pi x))$	9.0986452566e−03
22	0:2π	$x\sin(30x)\cos(x)$	$=-\frac{60}{899}\pi$
23	0:1	$4\pi^2 x\sin(20\pi x)\cos(2\pi x)$	$-\frac{20\pi}{99}$
24	0:2π	$e^{-x}\sin(10x)$	$\frac{1.0-e^{(-2\pi)}}{10.1}$

D in paper [28]). Numerical integration of singular functions (group B and group E in [28]) on coarse meshes remains a challenging problem whose discussion is beyond the scope of this paper.

Let us compute the error of numerical integration as

$$\|e\| = \frac{|I - I_a|}{|I|}, \tag{8}$$

where I is the exact integral and I_a is the approximate integral obtained by a chosen method. Table 2 shows the error (8) for the Gauss–Legendre rule, where we use exact values of the integrand $f(x)$ at the roots x_k of Legendre polynomials in each grid subinterval $[x_n, x_{n+1}]$, $n = 0, \ldots, N-1$. It also shows the error (8) for the GLI method and for the Simpson rule. The integration error for the above-mentioned methods is denoted in the table as e_{GR}, e_{GLI} and e_S, respectively. The number K of Gauss–Legendre abscissae is chosen as $K = 5$ to ensure that the accuracy of the Gaussian quadrature is far more superior than the accuracy of interpolation.[1] The function $f(x)$ is interpolated by a Lagrange polynomial of degree $p = 5$, so that the function values at six grid nodes are required for the interpolation, as it has been discussed in the previous section.

It can be seen from the table that for many test cases (problems 1–9, 12, 13, 24) the integration error for the GLI method is much smaller than the error for the Simpson rule. Meanwhile, there are several test cases where the GLI error is compatible or even larger than the error in the Simpson method. In particular, the integration error is very large in test cases 14–17 and 21–23 so that the results of integration on a coarse mesh cannot be acceptable by any standards. Thus our next goal is to discuss the issue of reliability of the numerical integration on a coarse mesh and to elaborate a criterion of mesh coarseness.

4.1. A coarse mesh problem

The asymptotical error estimates obtained in Section 3 say that a higher order GLI method should have better accuracy than the Simpson method if the mesh is refined. Hence, let us assume (for the sake of this discussion only) that an initial coarse mesh can be uniformly refined and compare the two methods of numerical integration on a sequence of refined meshes. Our criterion of mesh coarseness is then based on the applicability of asymptotic error estimates. Namely, if the error of the Simpson rule on a given mesh taken from the sequence of uniform refined meshes is smaller or compatible with that of the GLI method, we then conclude that the mesh is too coarse to provide us with a reliable integral value.

The above statement is illustrated by test case 16 in the Table 1. The function $f(x) = 1/(x+\delta)$, $\delta = 10^{-4}$, is shown in the interval $[0, 1]$ in Fig. 1a. It can be seen from the figure that the function has a narrow region of steep gradient ('a boundary layer') that appears because of the singularity at $x = -\delta$. Ideally, the integrand would be transformed to remove or reduce the strength of the singularity or an adapted grid would be employed in the problem where most grid nodes would be stationed

[1] Let us note that, while in the one-dimensional case the cost of the implementation of the 5-point Gauss–Legendre rule on coarse grids is negligible, the 3-point Gauss–Legendre rule should be recommended on two-dimensional grids.

Table 2
The error (8) for the exact 5-point Gauss–Legendre rule (e_{GR}), the GLI method with $p = 5$ (e_{GLI}) and the Simpson rule (e_S). The integration error is computed for the test cases of Table 1 on a coarse mesh with a fixed number N of mesh cells.

Testcase	N	e_{GR}	e_{GLI}	e_S
1	8	7.54016e−15	7.43722e−15	6.81175e−15
2	8	7.57727e−15	7.74066e−05	1.12661e−03
3	8	2.27457e−14	3.45983e−03	9.93023e−03
4	8	6.94122e−11	6.20026e−06	6.94349e−05
5	24	7.21645e−15	1.04631e−08	1.18628e−06
6	80	8.29322e−15	2.71808e−07	2.25232e−06
7	8	7.62426e−15	1.50800e−08	1.35382e−06
8	8	1.25732e−10	2.40093e−10	5.18887e−08
9	64	7.53649e−15	7.15278e−11	1.0989e−08
10	16	1.87464e−11	2.09435e−06	6.70322e−07
11	8	7.59659e−15	5.12995e−05	5.00607e−07
12	40	7.23632e−15	7.06803e−15	7.02111e−06
13	16	7.98924e−15	5.34135e−08	8.36774e−07
14	8	0.396073	14.9377	15.6668
15	8	1.9393e−02	1.80123	1.88942
16	8	2.83623e−01	42.5757	44.5519
17	8	3.85624e−01	3.99383	3.83861
18	8	2.65968e−03	4.65287e−01	4.9963e−01
19	80	3.31105e−04	3.46589e−01	3.94305e−01
20	8	4.42528e−07	6.24742e−03	3.38358e−03
21	8	6.46633	12.7436	12.4737
22	48	1.85629e−07	3.18952	5.02041
23	8	1.78086e−03	6.35748	6.33076
24	48	1.22018e−12	2.98335e−3	1.116795e−02

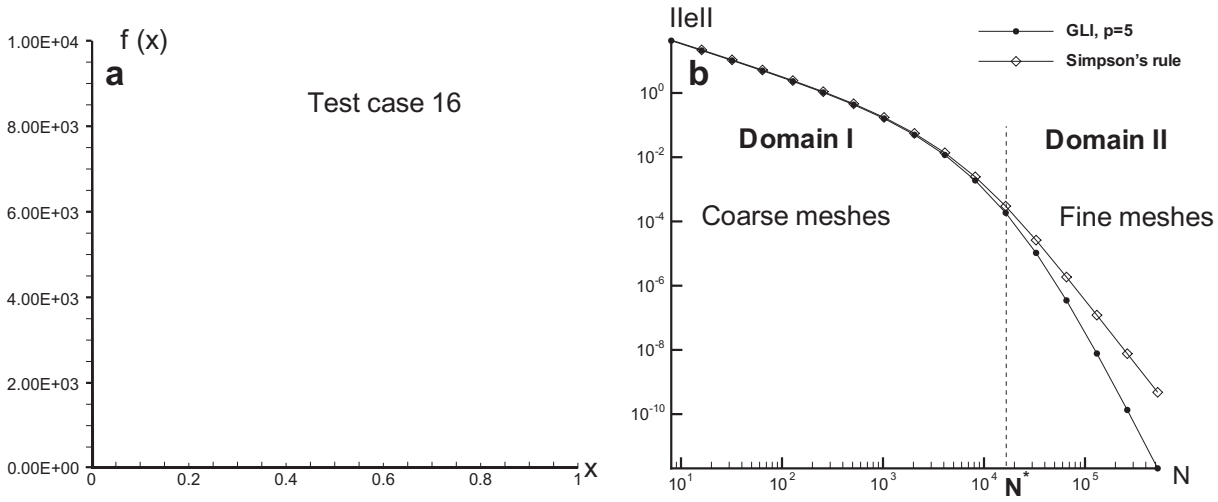

Fig. 1. The convergence of numerical integration methods for the function $f(x) = 1/(x + \delta)$, $\delta = 0.0001$. (a) The function $f(x)$ has a very narrow domain of the steep gradient ('a boundary layer') near the origin. (b) The asymptotic error estimates for the GLI method and the Simpson rule do not work on coarse meshes, unless the boundary layer of the width δ is resolved. The error of the GLI method is the same as that of the Simpson rule on coarse meshes where the number of mesh cells is $N < N^*$.

in the close vicinity of the origin. However we can only use a uniform mesh for the integration and cannot transform the integrand, as we assume that the analytical integrand is not available to us by the problem statement. As a result, the error of numerical integration is very large on the original coarse mesh as the mesh cannot capture the steep gradient region (see Table 2). Obviously, the value of the integral obtained on such a mesh cannot be considered reliable. Thus more grid points are required to resolve the integrand function near the point $x = 0$.

Let us uniformly refine the original mesh by halving each mesh cell. The refinement procedure is repeated several times and the integration error (8) on a sequence of nested meshes is shown in Fig. 1b for the GLI method with polynomial degree $p = 5$ and for the Simpson rule. The error in the figure is shown as a function of the number N of grid subintervals on a logarithmic scale. Since we use the polynomial degree $p = 5$ in the GLI method, we can expect a faster convergence rate in comparison with Simpson's rule. It can be seen from the figure, however, that both methods do not exhibit the expected convergence rate on the first several grids in the grid sequence.

The empirical convergence order q can be computed via

$$q = log_2\left(\frac{\|e\|_N}{\|e\|_{N+1}}\right),$$

where $\|e_N\|$ is the error (8) on the grid with N subintervals. The convergence rates for the test case 16 are presented in Table 3. The expressions q_{GLI} and q_S respectively denote the GLI method with $p = 5$ and the Simpson method. It can be seen from the table that the expected order $q = 6$ for the GLI method is achieved after intensive grid refinement only. The convergence rate of the GLI method is not getting better than the convergence of the Simpson method, unless a sufficient number of nodes have been inserted in the region of the large gradient. This occurs when the number of grid subintervals marked as N^* in Fig. 1b is achieved. This suggests that any grid in Domain 1 where $N < N^*$ should be considered as a coarse grid not providing a reliable value of the integral. The value of the integral computed by the Simpson rule is very inaccurate on coarse meshes and it is not possible to improve the accuracy of integration by employing a method with better asymptotic error estimates. In other words, the user will not benefit from implementing higher order polynomials in the GLI method on coarse meshes with $N < N^*$. Indeed, the convergence order is the same for the Simpson rule and the GLI method and the value of q is much lower than expected. Meanwhile any grid with $N > N^*$ (Domain 2 in Fig. 1b) is a fine mesh where numerical integration will give us reliable approximation.

A simple heuristic estimate of the number N^* can be obtained if some information about spatial heterogeneity of the function $f(x)$ is available in the problem. Namely, the width Δx of the domain where the function gradient is steep can be evaluated as $\Delta x \approx \delta$. Let us require that at least one grid node of the uniform grid should be inserted in the 'boundary layer' region for integrand approximation with acceptable accuracy. This implies that the value of N^* should be chosen as $N^* \approx \frac{1}{\Delta x} \sim 10^4$. It can be seen from Fig. 1b and from Table 3 that the above heuristic estimate of N^* is in a good agreement with the conclusions based on the knowledge of integration error.

The problem is further illustrated by consideration of test cases 14 and 21 in Table 1. Test case 14 presents the integrand $f(x) = (x + 0.01)^{-5}$ that again has a sharp gradient near the origin, while the integrand $f(x) = \sin(100\pi x)/(\pi x)$ in test case 21 is a rapidly oscillating function. In both cases the integration error on the initial coarse mesh is very large (see Table 2). The convergence history for these test cases is shown in Fig. 2a and b respectively. Again, it can be seen from the figure that the transition from coarse to fine meshes happens on meshes where number of mesh nodes is $N \approx N^*$. The number N^* is

Table 3
The convergence for the test case 16 in Table 1. The convergence order q of the GLI method with polynomial degree $p = 5$ (q_{GLI} in the table) and the Simpson method (q_S in the table). N is the number of grid subintervals.

N	16	32	64	128	256	512	1024	2048
q_{GLI}	1.018	1.032	1.055	1.094	1.157	1.261	1.428	1.688
q_S	1.018	1.031	1.053	1.090	1.151	1.250	1.408	1.651
N	4096	8192	16,384	32,768	65,536	131,072	262,144	524,288
q_{GLI}	2.082	2.638	3.351	4.151	4.910	5.494	5.842	6.043
q_S	2.009	2.487	3.032	3.515	3.816	3.945	3.985	3.996

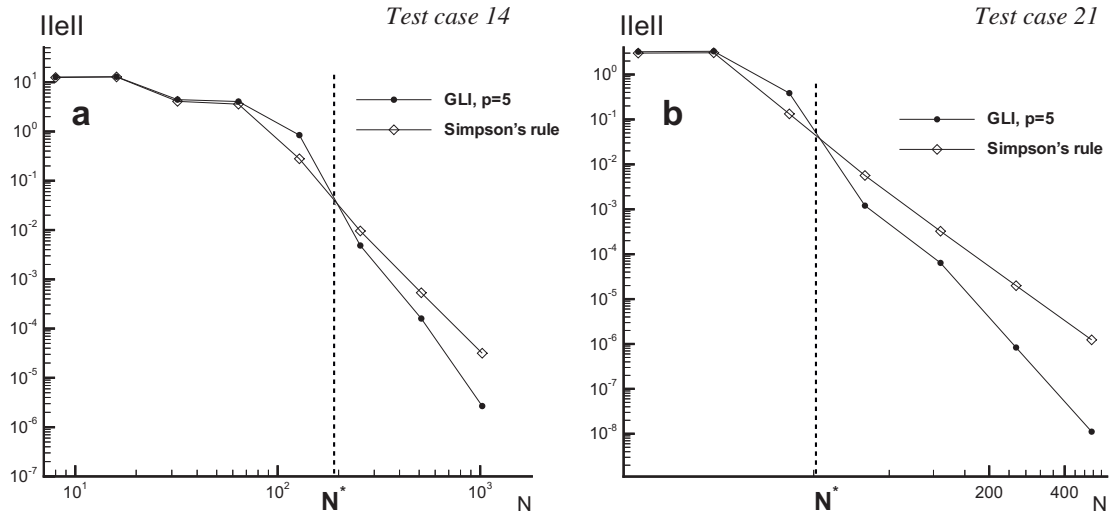

Fig. 2. Examples of the convergence rate for integration error (8) on coarse and fine meshes. The error is computed on a sequence of uniformly refined meshes for the GLI method with $p = 5$ and the Simpson rule. The convergence rate is shown for the integrand (a) $f(x) = (x + 0.01)^{-5}$ and (b) $f(x) = \sin(100\pi x)/(\pi x)$.

considered the minimum number of grid cells required to provide a reliable integral evaluate. While asymptotic error estimates hold on fine meshes with $N > N^*$, the convergence rate cannot be predicted in the domain of coarse meshes $N < N^*$ and a reliable value of the integral cannot be obtained.

It is worth noting here that, as the mesh coarseness is now evaluated in terms of the integration error rather than by the number of mesh nodes, its definition depends strongly on the integrand function. The number N^* obtained for the integrand in test case 14 is $N^*_{14} \approx 200$, while in test case 21 the transition to fine meshes occurs on the grid with the number $N^*_{21} \approx 50$ of grid subintervals. Thus a mesh considered as coarse for one integrand function can be a fine mesh for another integrand.

Going back to Table 2, we now see that test cases $1 - 9$,12, 13 and 24 allow one to use a mesh with a very small number of subintervals for accurate integral evaluation. Since the asymptotic error estimates hold on the mesh of $N = 8$ subintervals generated over the unit interval, the results of our discussion allow this mesh to be considered as a fine mesh for the above-mentioned test cases. This is not true for other test cases where the mesh should be further refined to provide an asymptotic convergence rate.

5. Reliability criterion in practical applications: Ecological examples

In this section we extend the previous conclusions based on conventional test cases to ecologically meaningful applications where a coarse mesh problem can be re-formulated as a problem of minimization of the number of traps used to obtain the density of pest insect population [24]. From the experimental viewpoint the conclusions made in the previous section suggest the following reliability criterion for computations made on a coarse mesh. Let us assume, for instance, that N_0 equidistant detectors (traps) have been installed in the domain $[0, 1]$ for data collection. Our approach to reliability of computations requires to evaluate the number N^* and to check if $N_0 > N^*$. In case the above condition holds we will benefit from using higher order polynomials in the GLI method, as the results of the GLI integration will be more accurate than the Simpson rule on a mesh with number N_0 of mesh nodes. On the contrary, if $N_0 < N^*$ we conclude that the number N_0 of traps is not sufficient to collect the required information about the integrand function and the integral value obtained for those sampled data will not be reliable.

Below we illustrate this approach by consideration of two 'real-life' test cases where the integrand $f(x)$ is the pest population density observed in ecological applications. The examples below have been taken from the work [24] where a more detailed discussion of the ecological issues related to the problem can be found. The integrand $f(x)$ has been obtained by numerical simulation of an ecological problem based on the consideration of a system of coupled time dependent diffusion–reaction equations (the Rosenzweig–MacArthur model) [22]. The distribution $f(x)$ is shown in Fig. 3 for different values of time and the diffusion coefficient in the problem.

As in our previous test cases we look at the convergence rate for each of the distributions of the pest population in order to conclude about the reliability of computations on the initial coarse grid of $N = 8$ grid cells. For the rest of this subsection we assume that the integration error is available to us. When the error is not available, the problem of calculating the threshold number N^* is briefly addressed in Section 5.1.

The convergence history for the pest population density $f(x)$ shown in Fig. 3a is presented in Fig. 4a. This test case clearly demonstrates the advantage of the GLI method with a higher order polynomial used for the interpolation, as the error of the method is smaller on any refined mesh. We conclude from the convergence graphs that the initial grid of $N_0 = 8$ subintervals is not a coarse mesh for the integrand in Fig. 3a and the results of integration on this grid are reliable as asymptotic error estimates can be applied.

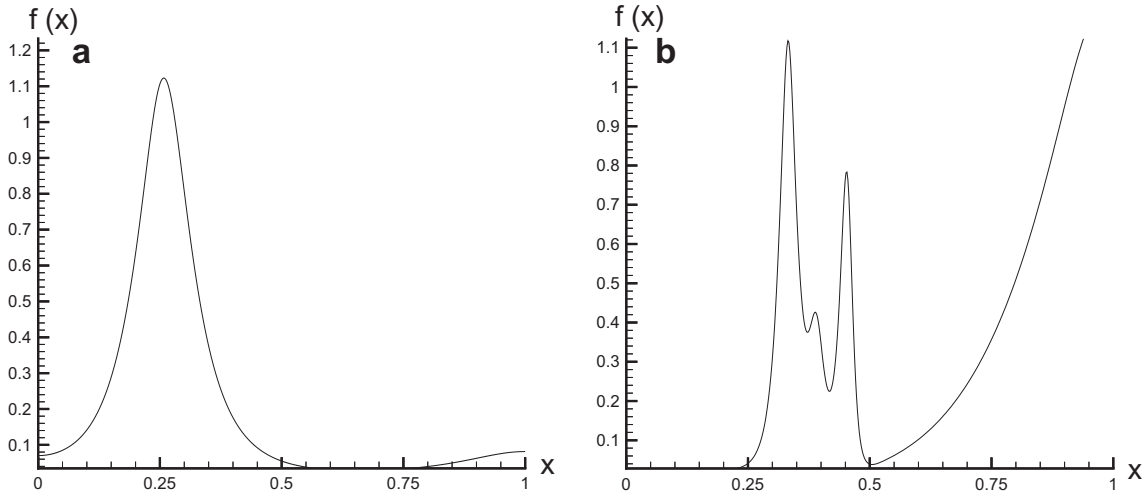

Fig. 3. Ecological test cases. Pest population density over space as predicted by the Rosenzweig–MacArthur model at different times t and for different values of the dimensionless diffusivity d: (a) for $d = 10^{-4}$ at $t = 50$ and (b) for $d = 10^{-5}$ at $t = 100$.

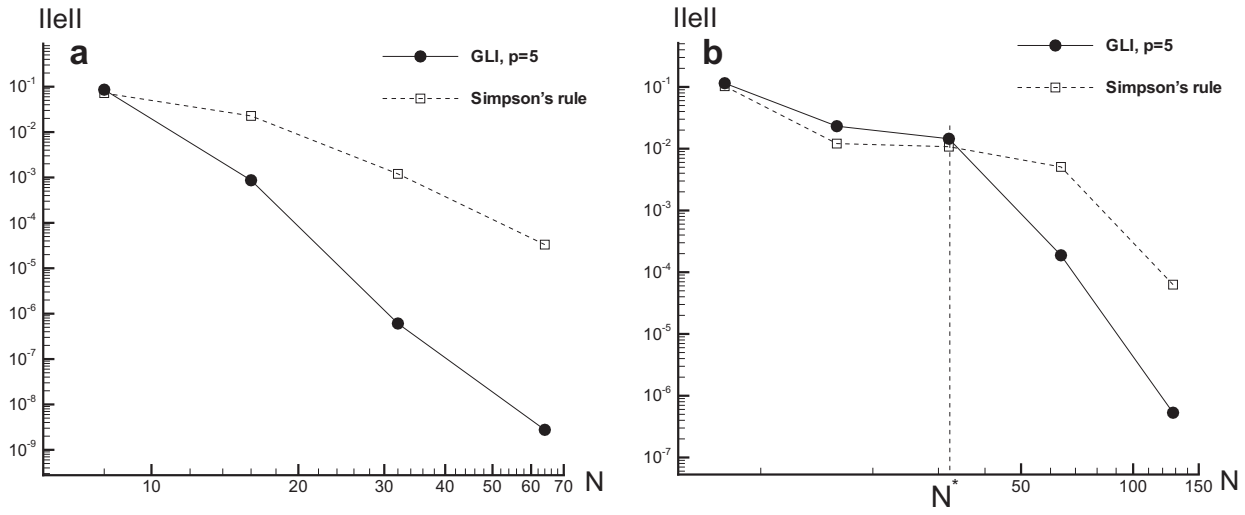

Fig. 4. The convergence history for ecological test cases shown in Fig. 3. The convergence rate is shown for the GLI method with $p = 5$ (solid line) and the Simpson rule (dashed line) on a logarithmic scale. (a) The convergence rate for a distribution of the pest population density in Fig. 3a. (b) The convergence rate for an oscillating distribution of the pest population density in Fig. 3b.

The convergence test for the pest population density of Fig. 3b is presented in Fig. 4b. It can be seen from the figure that a finer grid should now be generated to resolve the humps that are present in the solution. In case that we start from a coarse mesh of $N_0 = 8$ cells several cycles of uniform refinement are required to resolve solution peaks for a more complex pattern of the population density shown in Fig. 3b. From the consideration of the convergence graphs for the GLI with $p = 5$ and the Simpson method, it follows that a grid of $N^* \sim 40$ cells is sufficient to obtain a reliable integral value as asymptotic error estimates hold for both integration methods on grids with $N > N^*$. Hence, $N_0 = 8$ is not a sufficient number of traps per unit interval and more traps should be installed in order to provide an accurate integral value. At the same time, a counter-intuitive conclusion following from our study of Fig. 4b is that if it is not possible to increase the number of traps to reach the threshold value N^*, there is no need to increase the number N of traps at all. It can be seen from Fig. 4b that increasing the number of grid nodes in the domain of coarse meshes (that is, $N < N^*$) does not bring any gain in accuracy of integration. This conclusion is also confirmed by the results of test cases shown in Fig. 2, where the increase in accuracy is very small unless the grid is refined to the limit $N \sim N^*$.

5.1. Empirical error approximations

The above discussion has been based on the assumption that the integration error is available in the problem. This assumption is, of course, not true in practical applications, where the integration error is not available and heuristic estimates are conventionally used for the number of detectors/traps required for accurate numerical integration. Hence, our next goal is to demonstrate how a heuristic approach used in ecological applications can be checked against a more rigorous estimate of mesh coarseness considered in the paper. A heuristic estimate depends on a particular practical problem at hand. We consider below an example where a heuristic estimate based on the spatial heterogeneity of the population density distribution has been obtained in Ref. [24] for the Rosenzweig–MacArthur model.

Let Δx be a characteristic width of a spatial heterogeneity described by a given integrand (that is, the width of a single hump in the ecological examples above). It has been discussed in Ref. [24] that the threshold number N^* is defined by Δx and it can be evaluated as

$$N^* \approx k\frac{1}{\Delta x}, \tag{9}$$

$k \geqslant 1$ is a numerical coefficient that depends on the heterogeneity type, $k = 3$ for a single hump. In ecological applications the hump width Δx is, in turn, defined by the diffusion d and it has been found in work [25] that Δx can be evaluated as

$$\Delta x \approx \omega\sqrt{d}, \tag{10}$$

where the coefficient $\omega \sim 25$ (see [25] for the derivation of ω). Let us consider the pest population density distribution shown in Fig. 3b where the diffusion coefficient is $d = 10^{-5}$. The estimate (10) gives us the hump width $\Delta x \approx 0.08$, so that the number N^* can be evaluated from (9) as $N^* \approx 40$. It can be seen from Fig. 4b that the above heuristic estimate of mesh coarseness is in a good agreement with our estimate of the number N^* of mesh cells based on the knowledge of the integration error. Hence the conclusions of the previous subsection to increase the number of traps for the oscillating pattern of the density population are correct and should be taken into account when field measurements are carried out.

6. Concluding remarks

We have considered a problem of numerical integration in the case that an integrand function is defined at nodes of a coarse uniform mesh. Our study has originally been prompted by the needs of ecological monitoring and control where minimization of the number of experimental measurements made to accurately define the integrand remains an important problem. However a coarse mesh problem also arises in a wide range of practical and computational applications where in many cases only heuristic estimates of the number of mesh subintervals are available in the problem and criteria of the coarse mesh quality are still unclear (e.g., see [6]).

Our choice of the integration method has been dictated by the requirements of accuracy and flexibility, as conventional methods used on uniform grids are sensitive to the deviations in grid node locations and higher order polynomial approximation may suffer from the Runge phenomena of spurious oscillations. Hence we implemented a composite integration rule in the problem, where local polynomial reconstruction is used to integrate a given function in each grid subinterval. The method can therefore be used on non-uniform grids and the order of approximation can easily be controlled.

The accuracy of the GLI method depends on the degree of the polynomial used in the method formulation for solving the coarse mesh problem. Exploiting higher order polynomials in the GLI method should result in a smaller integration error in comparison with the error of the Simpson rule widely used in practical computations on uniform meshes. However, several test cases carried out in the paper demonstrate that the error of the GLI method is larger than the error of the Simpson rule. The analysis of those test cases results in a revised definition of a coarse mesh. Based on the convergence rate of the two methods a coarse mesh is now defined as a mesh where one cannot apply asymptotic error estimates to evaluate the integration error.

The findings of the paper can be summarized as follows:

1. The mesh coarseness is defined by the properties of the integrand, not by the number of grid subintervals, and conclusions about mesh coarseness based just on the given number N of grid cells can be misleading. The examples discussed in the paper demonstrate that the spatial heterogeneity of the integrand function should be taken into account when the question is considered whether the threshold number N^* of grid subintervals is sufficient for an accurate integration.
2. The number N^* is defined by the integration error. In practical applications, where the integration error is not available, the information about the typical scale of spatial heterogeneity should be secured to obtain a heuristic estimate of the threshold number N^* of detectors/traps in the problem.
3. The integration strategy should be dictated by the threshold number N^*:
   - Using higher order polynomials to approximate the integrand does not result in more accurate approximation on coarse meshes where the number of mesh cells is $N < N^*$. The test cases considered in the paper reveal that the integration error of a higher order GLI method remains approximately the same as the error of Simpson's rule on coarse meshes with $N < N^*$. A user will benefit from more accurate integration provided by a higher order method only if the number of mesh subintervals is increased to N^*. This conclusion is important as nowadays it is a widespread opinion in the computational community that higher order polynomial approximation may help one to obtain a more accurate answer even if an extremely coarse mesh is considered (e.g., see [8,23,29]).
   - Similarly, if the mesh cannot be refined to make the number of cells $N > N^*$ there is no need to refine it at all, as a larger number of mesh cells will not provide better accuracy. From a practical viewpoint this conclusion means that if the number N^* of traps is not available in field measurements, then one can consider the minimum number of traps without trying to increase it.
   - If the number N of mesh cells is $N > N^*$, an integration method with better asymptotic error estimates will provide more accurate integration. In particular, a GLI method discussed in the paper can be used for integration on grids with $N > N^*$ where employing higher order polynomials to approximate the integrand will result in a more accurate integral value in comparison with the Simpson method.

Our future work is to study multi-dimensional cases as we want to apply our approach to real-life ecological problems where the population size of a given species has to be restored based on measurements of the population density. Heuristic estimates obtained for common pest species distributions should be carefully investigated and checked against a more rigorous estimate N^* discussed in the paper for those test cases where the integrand function is available from a mathematical model. Also, we would like to note here that standard methods of numerical integration have to be revised when one has to deal with coarse meshes. A conventional approach to the design of methods of numerical integration is based on consideration of asymptotic error estimates in the form Ch^p, where h is the grid step size, p is the polynomial degree and the constant C is defined by the properties of the integrand. The main effort in derivation of methods for integration on fine meshes is focused on increasing the value p to obtain a faster convergence rate. Meanwhile, our experience with numerical integration on coarse meshes allows us to conclude that a careful investigation of the coefficient C is required in design of methods on coarse meshes where asymptotic error estimates do not work and the error depends mainly on the value of C on a given mesh. At the moment it is not clear what improvements can be done to reduce the error on coarse meshes with $N < N^*$ and this challenging task should become another topic of future work.

Appendix A. Algorithm for the GLI method

Below we briefly explain the details of practical implementation of the Gauss–Legendre formula with interpolation in order to illustrate the method itself as well as the definition of a local interpolation stencil that we use in the problem. The algorithm for the GLI method can be written as follows:

1. Initialization
   Generate a uniform computational grid $\mathcal{G}_{[a,b]}$ as a set of nodal coordinates $x_0 = a, x_{n+1} = x_n + h, n = 0, \ldots, N-1, h = (b-a)/N$. Define (or read from a database) data $f_n = f(x_n)$, $n = 0, \ldots, N$. Set the value of integral $I_a = 0$.
   Define the order p of a local Lagrange interpolant. Define the number K of abscissae in the Gauss–Legendre formula and compute the abscissae $\tilde{\xi}_k$ and weights $\tilde{w}_k$, $k = 1, 2, \ldots, K$ of the Gauss–Legendre quadrature over the interval $[-1, 1]$.
2. Computation of the integral I over the domain $[a, b]$
   Do for $0 \leqslant n \leqslant N-1$
   2.1. Consider grid subinterval $e_n = [x_n, x_{n+1}]$. Analyze the location of the left endpoint x_n and define a local uniform grid $\mathcal{G}_{[X_1,X_2]}$ (a local interpolation stencil). Define the local Lagrange interpolant $L_p^n(x)$ and use it in order to interpolate function values from the grid $\mathcal{G}_{[X_1,X_2]}$ to points ξ_k, $k = 1, 2, \ldots, K$, where $\xi_k \in e_n$ are the abscissae in the Gauss–Legendre formula considered in the subinterval e_n.
   2.2. *Do* for $1 \leqslant k \leqslant K$
   2.2.1. Compute $\xi_k = \frac{1}{2}(x_n + x_{n+1}) + \frac{1}{2}(x_{n+1} - x_n)\tilde{\xi}_k$. Compute $w_k = \frac{1}{2}(x_{n+1} - x_n)\tilde{w}_k$.
   2.2.2. Interpolate the integrand function at the point $\xi_k \in e_n$ as $f(\xi_k) = L_p^n(\xi_k)$.
   2.2.3 Apply the Gauss–Legendre rule on the subinterval $e_n = [x_n, x_{n+1}]$ to compute the integral I_a as $I_a = I_a + L_p^n(\xi_k)w_k$.
   Go to 2.2.1. for $k = k + 1$ (that is the next abscissa ξ_k on the grid subinterval e_n).
   EndDo

   Go to 2.1. for $n = n + 1$ (that is the next grid subinterval e_n where the entire set $\{\xi_k\} \in e_n$, $k = 1, \ldots, K$ will be recomputed).
   EndDo
3. Postprocessing
Define the exact value I of the integral (if available). Compute the error of integration.

The calculation of the weights and abscissae of the Gaussian quadrature formulas is a well-documented problem (e.g., see [15]). In our program we use the ALGLIB library [5] that exploits the algorithm in Ref. [15].

We now have to explain how to define a local interpolation stencil for the Lagrange interpolant in order to complete the algorithm of numerical integration above. For this purpose we need to generate the local uniform grid $\mathcal{G}_{[x_i,x_{i+p}]}$ that will consists of the nodes $x_i, x_{i+1}, \ldots, x_{i+p}$ of the computational mesh $\mathcal{G}_{[a,b]}$. For any subinterval $e_n = [x_n, x_{n+1}]$, $n = 0, \ldots, N-1$ the choice of the left endpoint x_i in the interpolation stencil $\mathcal{G}_{[x_i,x_{i+p}]}$ depends on the location of the point x_n and on polynomial degree p. Below we discuss the case $p = 5$, which is the highest polynomial degree we use in our computations.

Consider the grid cell $e_n = [x_n, x_{n+1}]$, $n = 0, \ldots, N-1$. We use the following local numeration of stencil nodes in our numerical integration computer program: $\mathcal{G}_{[\hat{x}_{i'},\hat{x}_{i'+p}]} = \bigcup_{i'=1}^{p+1}[\hat{x}_{i'}, \hat{x}_{i'+1}]$. Thus we have $i' = 1, \ldots, 6$ for polynomial degree $p = 5$, and the local interpolation stencil is defined for the subinterval e_n as follows:

1. If $n = 0$ (that is, $x_n = a$, the left boundary of the domain $[a, b]$), then the stencil points are $\hat{x}_{i'} = x_{n+i'-1}$.
2. If $n = 1$ (x_n is the node next to the left boundary of the domain $[a, b]$), $\hat{x}_{i'} = x_{n+i'-2}$.
3. If $n > 1$ and $n < N-1$ (x_n is an interior node of the domain $[a, b]$), $\hat{x}_{i'} = x_{n+i'-3}$.
4. If $n = N-2$ (x_{n+1} is the node next to the right boundary of the domain $[a, b]$), $\hat{x}_{i'} = x_{n+i'-4}$.
5. If $n = N-1$ (x_n is the node next to the right boundary of the domain $[a, b]$), $\hat{x}_{i'} = x_{n+i'-5}$.

The choice of an interpolation stencil is illustrated in Fig. 5. Stencil (I) in the figure captures points required for interpolation at the left boundary cell $e_0 = [x_0, x_1]$. Stencil (II) is used for interpolation at an interior grid cell e_n of the domain $[a, b]$, while stencil (*III*) is used for interpolation at the right boundary cell $e_{N-1} = [x_{N-1}, x_N]$. Once the local interpolation stencil has been generated, the values $f(\hat{x}_{i'})$ are used to construct the interpolant (3). It can be readily seen from Fig. 5 how to reduce the number of stencil points if we want to use a polynomial degree $p < 5$ for the Lagrange interpolation.

Let us note that the GLI method is compared in our computational code with the composite Simpson rule, where the conventional application of the composite Simpson rule requires introducing the the'half-integer' nodes. In our computer program we change the node numeration to consider the odd-numbered nodes instead of the half-integer ones, so that the composite Simpson rule applied on a grid of N subintervals will include the local 'stencils' $[x_0, x_1, x_2], \ldots, [x_{N-2}, x_{N-1}, x_N]$. Thus from a computational viewpoint the GLI method is in general more expensive than the composite Simpson rule

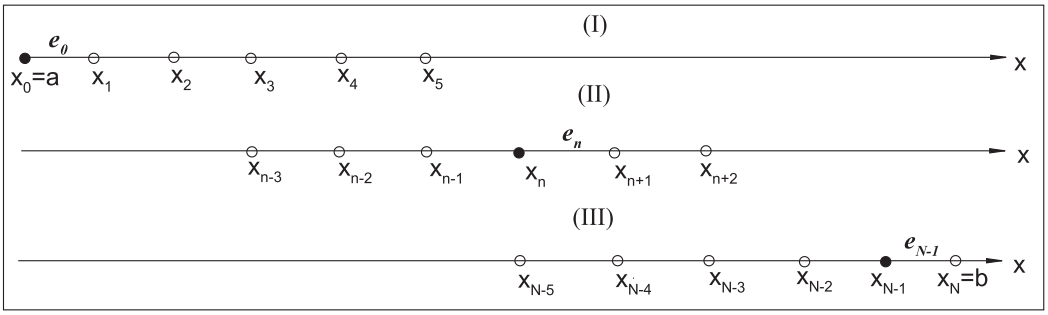

Fig. 5. Examples of a local interpolation stencil for Lagrange interpolation.

on a uniform grid. However, for numerical integration on a coarse mesh the cost of computation in the former case is quite low and it is easy to allocate computational resources required for the GLI integration, because the number N of grid cells is small.

References

[1] C.J. Alexander, J.M. Holland, L. Winder, C. Wooley, J.N. Perry, Performance of sampling strategies in the presence of known spatial patterns, Ann. Appl. Biol. 146 (2005) 361–370.
[2] H.L. Atkins, C.W. Shu, Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations, AIAA 96-1683, 1996.
[3] K.E. Atkinson, E. Venturino, Numerical evaluation of line integrals, SINUM 30 (1993) 882–888.
[4] R.P. Blackshaw, The annual leatherjacket survey in Northern Ireland, 1965–1982, and some factors affecting populations, Plant Path. 32 (1983) 345–349.
[5] S. Bochkanov, ALGLIB Project. <http://www.alglib.net/>.
[6] G.S. Buscaglia, E. Dari, Anisotropic mesh optimization and its application in adaptivity, Int. J. Numer. Meth. Eng. 40 (1997) 4119–4136.
[7] J.A. Byers, O. Anderbrant, J. Lofqvist, Effective attraction radius: a method for comparing species attractants and determining densities of flying insects, J. Chem. Ecol. 15 (1989) 749–765.
[8] B. Cockburn, G.E. Karniadakis, C.-W. Shu, The development of discontinuous Galerkin methods, Lect. Notes Comput. Sci. Eng. 11 (2000) 3–50.
[9] P.J. Davis, P. Rabinowitz, Methods of Numerical Integration, Academic Press, New York, 1975.
[10] S.M. Dunn, A. Constantinides, P.V. Moghe, Numerical Methods in Biomedical Engineering, Elsevier Academic Press, USA, 2006.
[11] H. Engels, Numerical Quadrature and Cubature, Academic Press, 1980.
[12] G. Evans, Practical Numerical Integration, Wiley, NY, 1993.
[13] P. Favati, G. Lotti, F. Romani, ALGORITHM 691. Improving QUADPACK automatic integration routines, ACM Trans. Math. Softw. 17 (1991) 218–232.
[14] W. Gander, W. Gautschi, Adaptive quadrature – revisited, BIT 40 (2000) 84–101.
[15] W. Gautschi, Orthogonal Polynomials: Computation and Approximation, Oxford University Press, Oxford, 2004.
[16] T. Hasegawa, S. Hibino, Y. Hosoda, I. Ninomiya, An extended doubly-adaptive quadrature method based on the combination of the Ninomiya and the FLR schemes, Numer. Algor. 45 (2007) 101–112.
[17] J.M. Holland, J.N. Perry, L. Winder, The within-field spatial and temporal distribution of arthropods in winter wheat, Bull. Ent. Res. 89 (1999) 499–513.
[18] D.K. Kahaner, Computation of numerical quadrature formulas, in: J.R. Rice (Ed.), Mathematical Software, Academic, Orlando, FL, 1971, pp. 229–259.
[19] S. Karlin, Best quadrature formulas and splines, J. Approx. Theory 4 (1971) 59–90.
[20] S.X. Liao, M. Pawlak, On image analysis by moments, IEEE Trans. Pattern Anal. Mach. Intell. 18 (3) (1996) 254–266.
[21] H. Malchow, S.V. Petrovskii, E. Venturino, Spatiotemporal Patterns in Ecology and Epidemiology: Theory, Models, and Simulations, Chapman & Hall/CRC Press, London, 2008.
[22] J.D. Murray, Mathematical Biology, Springer, Berlin, Germany, 1989.
[23] C. Ollivier-Gooch, M. Van Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation, J. Comput. Phys. 181 (2002) 729–752.
[24] N.B. Petrovskaya, S.V. Petrovskii, The coarse mesh problem in ecological monitoring, Proc. Roy. Soc. A 466 (2010) 2933–2953.
[25] S.V. Petrovskii, B.-L. Li, H. Malchow, Quantification of the spatial aspect of chaotic dynamics in biological and chemical systems, Bull. Math. Biol. 65 (2003) 425–446.
[26] R. Piessens, E. deDoncker-Kapenga, E. Uberhuber, D. Kahaner, QUADPACK: A Subroutine Package for Automatic Integration, Springer-Verlag, Berlin, 1983.
[27] V.M. Stern, Economic thresholds, Ann. Rev. Entomol. 18 (1973) 259–280.
[28] A. Venter, D.P. Laurie, A doubly adaptive integration algorithm using stratified rules, BIT 42 (2002) 183–193.
[29] Z.J. Wang, High-order methods for the Euler and Navier–Stokes equations on unstructured grids, Prog. Aerosp. Sci. 43 (2007) 1–41.
[30] L. Yaroslavsky, A. Moreno, J. Campos, Frequency responses and resolving power of numerical integration of sampled data, Opt. Exp. 13 (8) (2005) 2892–2905.