

On Complexity of Adaptive Splines

Yuri K. Demjanovich
Saint-Petersburg State University
Russia

Received: June 25, 2020. Revised: August 31, 2020. 2nd Revised: September 18, 2020.
Accepted: September 21, 2020. Published: September 22, 2020.

Abstract—The paper discusses various methods of adaptive spline approximations for the flow of function values. It is considered an adaptive compression algorithm, which, for a priori given $\varepsilon > 0$, has the properties 1) the complexity of the algorithm is proportional to the length of the original flow, 2) by the piecewise linear interpolation of the compression result, it is possible to restore the original flow with an accuracy of ε , 3) the compression result is close to optimal and has $O(M)$ of arithmetic operations. The effectiveness of this approach is demonstrated on rapidly changing initial flows of numerical information in the digital experiment. In addition, the paper presents an exact two-sided estimate for the number $O(M^2)$ of arithmetic operations for the optimal solution of the problem of compressing an informational numerical flow of length M with the possibility of recovering this flow with a predetermined accuracy. Provided that the original flow is convex, a compression algorithm is developed with an accurate two-sided estimate of the number $O(M \log_2 M)$ and with the possibility of recovery with a prescribed accuracy.

Keywords—spline approximation, computational complexity, adaptive grid, algorithms of enlargement.

I. INTRODUCTION

HE splines are a widespread apparatus for processing numerical information flows (see [1] – [33]). However, the needs associated with the constant increase in the volume of the numerical information require the further development of this apparatus. In this regard, we mention the development of splines that satisfy additional conditions. The various types of splines are considered, including sign-defined splines, the splines preserving the prescribed shape, and adaptive splines, etc. (see [13], [16], [25], [26], [31]). The complexity of the compression of numerical flows is a basic problem for signal processing. Standard compression of flow in classical wavelet algorithms is performed by removing components with odd knots (see [10]). In this case, there is no reason to hope for a qualitative approximation of the initial flow. Paper [4] discusses the computer complexity for the interpolation of n

data by the splines of order m with the result $m^2 n/4$ arithmetical operations. In paper [6], J.H.Friedman uses the splines for the modelling of the prediction in statistics. The method (named MARS) for flexible regression analysis of multivariate data is presented. The complexity of the model building algorithm depends significantly on the nature of the input data. Paper [7] is devoted to B-spline complexity in the case of a uniform grid. The complexity of the approximation is proportional to the number of data. The authors of paper [21] consider strict conditions that must obey the boundary curves of smooth finite elements in the case of quadrangular grids. Fast polynomial spline with prescribed properties is represented in paper [23]. In the paper authors show that the complexity of calculating the unknown derivatives is a linear function of the length of the initial data flow. Usage of the cubic splines as the apparatus for application to construction of adaptive linear filter is discussed in paper [27]. An application of parallel technology CUDA with usage of B-splines is demonstrated in paper [29]. The authors of all enumerated papers have a tendency to optimize spline processing. The natural source of optimization is the adaptive processing for the initial data flow.

The purpose of this work is to develop an adaptive compression algorithm that, for a priori given $\varepsilon > 0$, has the properties 1) the complexity of the algorithm is proportional to the length of the original flow, 2) by the piecewise linear interpolation of the compression result, it is possible to restore the original flow with an accuracy of ε , 3) for rapidly changing smooth flows, the compression result is close to optimal. In addition, the paper presents an exact two-sided estimate for the number $O(M^2)$ of arithmetic operations for the optimal solution of the problem of compressing an informational numerical flow of length M with the possibility of recovering this flow with a predetermined accuracy. Provided that the original flow is convex, a compression algorithm is developed with an accurate two-sided estimate of the number $O(M \log_2 M)$ and with the possibility of recovery with a prescribed accuracy.

In the case of an analog signal the smoothness means the continuous differentiability of the function that generates the flow. For a discrete flow that means the boundedness of the difference ratios. In this work, the complexity of the algorithms is determined by the number of calculations of the initial flow, by the number of additive and multiplicative operations, and by the number of comparison operations. All

of these characteristics are assessed separately. They are often referred to as arithmetic operations. As a result we obtain a small numerical flow which can be used for piece-wise linear restoration $\tilde{u}(\xi_j)$ of the initial flow $u(\xi_j)$ with a predetermined precision $\varepsilon > 0$, where ξ_j are knots of initial fine grid. The best solution of this problem exists and gives optimal compression. Generally speaking, this solution is required of $O(M^2)$ arithmetic operations. This is due to the fact that an attempt to replace "chord" $\tilde{u}(t)$ to another "chord" requires finding of all differences $\tilde{u}(\xi_j) - u(\xi_j)$ in knots ξ_j of initial fine grid (Section V contains more details and accurate estimates). Thus, for $M = 10^9$ the number of arithmetic operations can be overwhelming. It is possible to improve the algorithm, if the initial data has the convex property. In this case, it is sufficient $O(M \log_2 M)$ arithmetic operations (see Section VI for the exact wording). We propose a compression algorithm, requiring $O(M)$ arithmetic operations. Of course, in this case it is necessary to abandon the above-mentioned optimal compression. We obtain the resulting compressed flow on the grid which depends on the initial flow essentially. Therefore, this grid is called the adaptive grid. The relevant approximating spline is called the adaptive spline. The effectiveness of this approach is demonstrated on rapidly changing initial flows of numerical information (see digital experiment in Section VII).

The idea behind this work is to replace the computation of absolute values $|\tilde{u}(\xi_j) - u(\xi_j)|$ by calculating their evaluations. The evaluations can include derivatives or corresponding difference relations. The evaluations differ significantly from the mentioned absolute value because they have the property of a monotonic increase with an increase of the considered interval. Therefore it sufficient to find the maximum of two numbers instead of multiple calculations of mentioned absolute value for the next interval. As a result, the number of arithmetic operations is substantially reduced. Instead of $O(M^2)$ we get $O(M)$ arithmetic operations.

II. SOME AUXILIARY ASSERTIONS

A. Adaptive grid

Consider a positive continuous function $f(t)$,

$$f \in C[a, b], \quad f(t) > 0 \quad \forall t \in [a, b]. \quad (1)$$

Let ε be a positive value. We suppose to discuss a grid

$$\tilde{X}(f, \varepsilon): a = \tilde{x}_0 < \tilde{x}_1 < \dots < \tilde{x}_K < \tilde{x}_{K+1} = b \quad (2)$$

such that

$$\max_{t \in [\tilde{x}_i, \tilde{x}_{i+1}]} f(t) (\tilde{x}_{i+1} - \tilde{x}_i) = \varepsilon \quad \forall i \in \{0, 1, \dots, K-1\}, \quad (3)$$

$$\max_{t \in [\tilde{x}_K, \tilde{x}_{K+1}]} f(t) (\tilde{x}_{K+1} - \tilde{x}_K) \leq \varepsilon. \quad (4)$$

Grid (2)–(4) is named *an adaptive grid for the function f*.

The next assertion holds.

Lemma 1. *If relations (1) are right, then for arbitrary*

$$\varepsilon \in (0, \varepsilon_0), \quad \varepsilon_0 = (b-a) \max_{t \in [a, b]} f(t), \quad (5)$$

a natural number $K = K(f, \varepsilon)$ and a grid

$$\tilde{X}(f, \varepsilon) = \{\tilde{x}_i(f, \varepsilon)\}_{i \in \{0, 1, \dots, K, K+1\}}$$

exist such that the properties (3)-(4) are fulfilled. The number $K = K(f, \varepsilon)$ is unique.

Proof. The lemma is proved by mathematical induction over the number of knots.

I. The induction base is set as follows. Consider a function $\varphi_0(\tau) = \max_{t \in [\tilde{x}_0, \tau]} f(t) (\tau - \tilde{x}_0)$. Let the variable τ increase from $a = \tilde{x}_0$ to b . Then, in view of the assumptions (1) the function $\varphi_0(\tau)$ strictly increases from 0 to $\max_{t \in [a, b]} f(t) (b-a)$. By condition (5) the unique point $\tau_1 \in [a, b]$ exists that $\max_{t \in [a, \tau_1]} f(t) (\tau_1 - a) = \varepsilon$. If we put $\tilde{x}_1 = \tau_1$ in the last formula we have relation (3) for $i = 0$. The induction base is set.

II. Now let us make an induction transition. We suppose that knots $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_s$ of the grid \tilde{X} have been defined, and relations (3) hold for $i = 0, 1, 2, \dots, s-1$. If $\tilde{x}_s = b$, then we put $K = s-1$. By the induction hypothesis, relation (3) holds for $i=s-1$, so $\max_{t \in [\tilde{x}_{s-1}, \tilde{x}_s]} f(t) (\tilde{x}_s - \tilde{x}_{s-1}) = \varepsilon$. Replacing here s on $K+1$, we see that formula (4) is valid. In this case the construction of the grid $\tilde{X}(f, \varepsilon)$ satisfying the conditions (3)–(4), is completed.

Otherwise, if $\tilde{x}_s < b$, the construction of the grid continues. Consider a function $\varphi_s(\tau) = \max_{t \in [\tilde{x}_s, \tau]} f(t) (\tau - \tilde{x}_s)$. If τ changes from \tilde{x}_s to b then the function $\varphi_s(\tau)$ increases from 0 to $m_s = \max_{t \in [\tilde{x}_s, b]} f(t) (b - \tilde{x}_s)$. In the case under consideration, two options are possible: 1) $\varepsilon < m_s$ and 2) $m_s \leq \varepsilon$. Let us consider each of them.

1) If $\varepsilon < m_s$ then there is a unique number τ_{s+1} , $\tau_{s+1} < b$, such that $\varphi_s(\tau_{s+1}) = \varepsilon$. Setting $\tilde{x}_{s+1} = \tau_{s+1}$ from the last formula we find

$$\varphi_s(\tilde{x}_{s+1}) = \varepsilon \iff \max_{t \in [\tilde{x}_s, \tilde{x}_{s+1}]} f(t) (\tilde{x}_{s+1} - \tilde{x}_s) = \varepsilon,$$

so relation (3) holds for $i=s$. In this case the step of the induction ends.

2) If $m_s \leq \varepsilon$ then we put $K = s$, $\tilde{x}_{s+1} = b$. Insofar as

$$m_s \leq \varepsilon \iff \max_{t \in [\tilde{x}_K, \tilde{x}_{K+1}]} f(t) (b - \tilde{x}_s) \leq \varepsilon,$$

then replacing s with K , we arrive at inequality (4). This completes the induction transition.

Thus, the induction transition is complete on the whole.

By relation (3) we have

$$\tilde{x}_{i+1} - \tilde{x}_i = \frac{\varepsilon}{\max_{t \in [\tilde{x}_i, \tilde{x}_{i+1}]} f(t)} \geq \frac{\varepsilon}{\max_{t \in [a, b]} f(t)} \quad \forall i \in \{0, 1, \dots, K-1\}.$$

Therefore we can conclude that the discussed process is finite. The Lemma is completely proved.

It is possible to discuss an initial fine grid

$$\bar{E}: \dots < \xi_{-2} < \xi_{-1} < \xi_0 < \xi_1 < \xi_2 < \dots$$

and consider the function $f(t)$ defined on a grid segment $[[a, b]] = \{a = \xi_0, \xi_1, \dots, \xi_{M+1} = b\}$.

It is evident in this case there is analog of the Lemma 1, in which $\tilde{X} \subset [[a, b]]$, and instead of condition (3) it needs to discuss the condition

$$\max_{t \in [\tilde{x}_i, \tilde{x}_{i+1}]} f(t) (\tilde{x}_{i+1} - \tilde{x}_i) \leq \varepsilon <$$

$$< \max_{t \in [\tilde{x}_{i+1}, \tilde{x}_{i+2}]} f(t) (\tilde{x}_{i+2} - \tilde{x}_{i+1})$$

$$\forall i \in \{0, 1, \dots, K-1\}.$$

It is clear to see that the next assertions are right.

Lemma 2. *If conditions (1) - (4) are true and $\varepsilon \rightarrow +0$*

then the integer function $K(f, \varepsilon)$ increases. The next relations

$$\lim_{\varepsilon \rightarrow 0} \max_{s \in \{0, 1, \dots, K\}} (\tilde{x}_{s+1} - \tilde{x}_s) = 0, \quad \lim_{\varepsilon \rightarrow 0} K(f, \varepsilon) = +\infty$$

are right.

Lemma 3. Under conditions (1) -- (4) the relation

$$\lim_{\varepsilon' \rightarrow 0} K(f, \varepsilon') \varepsilon' = \int_a^b f(t) dt \leq K(f, \varepsilon) + 1 \quad (6)$$

is fulfilled.

Summing (3) -- (4), we have

$$\begin{aligned} K\varepsilon &= \sum_{s=0}^{K-1} \max_{t \in [\tilde{x}_s, \tilde{x}_{s+1}]} f(t) (\tilde{x}_{s+1} - \tilde{x}_s) \leq \\ &\leq \sum_{s=0}^K \max_{t \in [\tilde{x}_s, \tilde{x}_{s+1}]} f(t) (\tilde{x}_{s+1} - \tilde{x}_s). \end{aligned} \quad (7)$$

B. Digital complexity

It is easy to see that the proof of Lemma 1 is actually an algorithm for construction of the grid \tilde{X} . Consider the question of the complexity of the calculations in this algorithm.

Let $N_{add} = N_{add}(f)$ and $N_{mul} = N_{mul}(f)$ be a number of additive and a number of multiplicative operations accordingly, as well as $N(f)$ and $N_{comp} = N_{comp}(f)$ be a number of calculations of the function $f(t)$ and a number of comparisons.

Lemma 4. The algorithm for the construction of the adaptive grid $\tilde{X}(f, \varepsilon)$ has the next properties

$$\begin{aligned} N(f) &= N_{add} = N_{mul} = K + M + 2, \\ N_{comp} &= 2(K + M + 2). \end{aligned} \quad (8)$$

Proof. The resulting grid has the form

$$\begin{aligned} \tilde{X} &= \tilde{X}(f, \varepsilon): \{a = \tilde{x}_0 < \tilde{x}_1 < \dots < \tilde{x}_K < \tilde{x}_{K+1} = b\} = \\ &= \{a = \tilde{x}_0 = \xi_0 < \tilde{x}_1 = \xi_{p_1} < \dots < \tilde{x}_K = \xi_{p_K} < \\ &\tilde{x}_{K+1} = \xi_{p_{K+1}} = b\}, \end{aligned} \quad (9)$$

where $p_{K+1} = M + 1$. At the s -th step of this algorithm, we move from the knot \tilde{x}_s to the knot \tilde{x}_{s+1} .

Suppose that $\tilde{x}_s = \xi_{p_s}$, $\tilde{x}_{s+1} = \xi_{p_{s+1}}$. It is not difficult to see that with the mentioned transition it is required

1) to calculate $p_{s+1} - p_s + 1$ times the function $f(t)$ (at points $\xi_{p_s+1}, \xi_{p_s+2}, \dots, \xi_{p_{s+1}+1}$);

2) to find the maximum of two numbers $p_{s+1} - p_s + 1$ times, namely find the next maximums

$$\begin{aligned} \mu_1 &= \max\{f(\xi_{p_s}), f(\xi_{p_{s+1}})\}, \mu_2 = \max\{\mu_1, f(\xi_{p_{s+2}})\}, \dots, \\ \mu_{p_{s+1} - p_s + 1} &= \max\{\mu_{p_{s+1} - p_s}, f(\xi_{p_{s+1}+1})\} = \max_{t \in [\tilde{x}_s, \tilde{x}_{s+1}]} f(t); \end{aligned}$$

3) to execute $p_{s+1} - p_s + 1$ additive operations, namely find the next differences

$$\begin{aligned} \sigma_1 &= \xi_{p_{s+1}} - \tilde{x}_s, \sigma_2 = \xi_{p_{s+2}} - \tilde{x}_s, \dots, \sigma_{p_{s+1} - p_s + 1} \\ &= \xi_{p_{s+1}+1} - \tilde{x}_s; \end{aligned}$$

4) to execute $p_{s+1} - p_s + 1$ multiplicative operations, namely find the following products

$$\begin{aligned} \theta_1 &= \mu_1 \sigma_1, \theta_2 = \mu_2 \sigma_2, \dots, \theta_{p_{s+1} - p_s + 1} = \\ &= \mu_{p_{s+1} - p_s + 1} \sigma_{p_{s+1} - p_s + 1} = \\ &= \max_{t \in [\tilde{x}_s, \tilde{x}_{s+1}]} f(t) (\tilde{x}_{s+1} - \tilde{x}_s); \end{aligned}$$

5) to make $p_{s+1} - p_s + 1$ comparisons with ε , namely, to obtain the following inequalities

$$\theta_1 \leq \varepsilon, \theta_2 \leq \varepsilon, \dots, \theta_{p_{s+1} - p_s} \leq \varepsilon, \theta_{p_{s+1} - p_s + 1} > \varepsilon$$

Since parameter s should be changed from 0 to K , then the total number of $N(f)$ calculations of the function

f is $N(f) = \sum_{s=0}^K (p_{s+1} - p_s + 1) = K + 1 + p_{K+1} - p_0$. Since, in accordance with (2) and (9), $p_0 = 0$, $p_{K+1} = M + 1$, then as a result, we get $N(f) = K + M + 2$. The same way we find the number $N_{add} = K + M + 2$ additive and the number $N_{mul} = K + M + 2$ multiplicative operations, as well as the number N_{comp} comparisons. So we get (8). This completes the proof.

Remark. Note that counting the number of operations is a very thankless task, because the result depends on the computing system and the programming style. For example, if the calculations in question are not performed on the remote computational modules, but on one module, then the estimate $N(f)$ can be improved. Really, there is no need to calculate the value $f(t)$ twice in knots of the form $\xi_{p_s+1}, \xi_{p_{s+1}+1}$ (in the segment $[\xi_{p_{s+1}}, \xi_{p_{s+2}}]$ you can use the value $f(\xi_{p_{s+1}+1})$, which is calculated in the previous stage of the calculation). In this case, the estimate $N(f)$ takes the form $N(f) = M + 1$. In this work, such subtleties are not considered. It is important for us that the complexity of the algorithm for obtaining the adaptive grid is proportional to the length of the initial data flow.

C. Pseudo-equidistant grid

Let J_K be a set $\{0, 1, 2, \dots, K\}$. The subset

$$\bar{X}: a = \bar{x}_0 < \bar{x}_1 < \dots < \bar{x}_{N+1} = b \quad (10)$$

of the grid segment $[[a, b]]$ is called *pseudo-equidistant grid* with width $h > 0$, if the next relations hold

$$\bar{x}_{j+1} - \bar{x}_j - 2\tau \leq h < \bar{x}_{j+1} - \bar{x}_j + 2\tau \quad j \in J_N, \quad (11)$$

where $\tau = \max_{j \in J_M} (\xi_{j+1} - \xi_j)$.

Suppose that the condition

$$2\tau \leq h < b - a \quad (12)$$

is fulfilled.

Let q be a real value. The expression $[q]$ means an integer number k_1 with the property $0 \leq q - k_1 < 1$. Analogously $[q]$ means an integer k_2 with the property $0 \leq k_2 - q < 1$.

For $j \in J_N$ we find $s \in J_M$ such that the inequality

$$\xi_s \leq jh < \xi_{s+1} \quad (13)$$

is right. By supposition (12) the unique number $s = s(j)$ exists. Let us discuss

$$\bar{x}_j = \xi_{s(j)} \quad \forall j \in J_N. \quad (14)$$

Lemma 5. The next relation holds

$$h - 2\tau < \bar{x}_{j+1} - \bar{x}_j \leq h + 2\tau. \quad (15)$$

Proof. We assume that

$$\xi_s \leq jh < \xi_{s+1}, \quad \xi_p \leq (j+1)h < \xi_{p+1}.$$

By supposition (12) we have $s < p$. If we put

$$\eta = jh - \xi_s, \quad \delta = (j+1)h - \xi_p$$

then we have

$$\xi_s = jh - \eta, \quad \xi_p = (j+1)h - \delta. \quad (16)$$

By (16) we deduce

$$0 \leq \eta < \xi_{s+1} - \xi_s, \quad 0 \leq \delta < \xi_{p+1} - \xi_p. \quad (17)$$

According to formulas (13) -- (14) we define

$$\bar{x}_j = \xi_s, \quad \bar{x}_{j+1} = \xi_p.$$

Thus we have

$$\bar{x}_{j+1} - \bar{x}_j = \xi_p - \xi_s = h - \delta + \eta. \quad (18)$$

By relations (18) and (17) we obtain (15). This completes the proof.

Relation (11) follows from (15), therefore the obtained grid is pseudo-equidistant.

Remark. If the initial grid is equidistant,

$$\xi_s = s\tau, \quad \tau = \frac{b-a}{M+1}, \quad s = 0, 1, 2, \dots, M+1,$$

then inequality (13) has the form $s\tau \leq jh < (s+1)\tau$. The last one is equivalent to the relation $s \leq \frac{jh}{\tau} < s+1$. Therefore we can put $s = \lfloor j \frac{h}{\tau} \rfloor$. Thus we have $\bar{x}_j = \xi_{\lfloor j \frac{h}{\tau} \rfloor}$.

Let $\varepsilon > 0$ be a positive value. We suppose that

$$N = \left\lceil \frac{\|f\|_{C[a,b]}}{\varepsilon} (b-a) \right\rceil > 3. \quad (19)$$

Then

$$\frac{\|f\|_{C[a,b]}}{\varepsilon} (b-a) \leq N < \frac{\|f\|_{C[a,b]}}{\varepsilon} (b-a) + 1. \quad (20)$$

The last inequality is equivalent to the relation

$$\begin{aligned} \|f\|_{C[a,b]} (b-a) &\leq N\varepsilon \\ &< \|f\|_{C[a,b]} (b-a) + \varepsilon \end{aligned} \quad (21)$$

We suppose that

$$\frac{b-a}{N+1} > 4\tau. \quad (22)$$

Choosing the value h according to the formula

$$h = \frac{b-a}{N+1} - 2\tau, \quad (23)$$

We see that by conditions (19) – (23) relation (12) is fulfilled. By (20) and (23) we have $N+1 > \|f\|_{C[a,b]}/\varepsilon$ and

$$h + 2\tau < \varepsilon \|f\|_{C[a,b]} \quad (24)$$

By (24) we have

$$\|f\|_{C[a,b]} (\bar{x}_{s+1} - \bar{x}_s) \leq \|f\|_{C[a,b]} (h + 2\tau) \leq \varepsilon. \quad (25)$$

By (25) we obtain

$$\max_{t \in [\bar{x}_s, \bar{x}_{s+1}]} f(t) (\bar{x}_{s+1} - \bar{x}_s) \leq \varepsilon \quad \forall s \in \mathbb{N}. \quad (26)$$

The previous arguments prove the following statement.

Theorem 1. *If relations (19), (22) are right then the pseudo-equidistant grid (10) exists and properties (21), (26) are fulfilled.*

D. Some assertions

Lemma 6. *If conditions (1)–(4), (19)–(23) are right then the next inequality is fulfilled:*

$$\begin{aligned} &\frac{\sum_{s=0}^K \max_{t \in [\bar{x}_s, \bar{x}_{s+1}]} f(t) (\bar{x}_{s+1} - \bar{x}_s) - \varepsilon}{(b-a) \max_{t \in [a,b]} f(t) + \varepsilon} \leq \frac{K(f, \varepsilon)}{N(f, \varepsilon)} \\ &\leq \frac{\sum_{s=0}^K \max_{t \in [\bar{x}_s, \bar{x}_{s+1}]} f(t) (\bar{x}_{s+1} - \bar{x}_s)}{(b-a) \max_{t \in [a,b]} f(t)}. \end{aligned} \quad (27)$$

Proof. By (7) we have

$$\sum_{s=0}^K \max_{t \in [\bar{x}_s, \bar{x}_{s+1}]} f(t) (\bar{x}_{s+1} - \bar{x}_s) - \varepsilon \leq K\varepsilon =$$

$$\sum_{s=0}^{K-1} \max_{t \in [\bar{x}_s, \bar{x}_{s+1}]} f(t) (\bar{x}_{s+1} - \bar{x}_s). \quad (28)$$

By (21) and (28) we obtain (26).

Theorem 2. *If the conditions of Lemma 5 are true then the relation*

$$\lim_{\varepsilon \rightarrow 0} \frac{K(f, \varepsilon)}{N(f, \varepsilon)} = \frac{1}{b-a} \frac{\int_a^b f(t) dt}{\max_{t \in [a,b]} f(t)} \quad (29)$$

is right.

Proof. Passing to the limit in (27) under condition $\varepsilon \rightarrow 0$, we obtain relation (29).

III. ON QUANTITY OF KNOTS FOR THE ADAPTIVE GRID

Consider a grid

$$\hat{X}: a = \hat{x}_0 < \hat{x}_1 < \dots < \hat{x}_K < \hat{x}_{K+1} = b. \quad (30)$$

Suppose $u \in C[a, b]$. Let $\tilde{u}(t)$ be a piecewise linear function

$$\tilde{u}(t) = u(\hat{x}_i) + \frac{u(\hat{x}_{i+1}) - u(\hat{x}_i)}{\hat{x}_{i+1} - \hat{x}_i} (t - \hat{x}_i) \quad t \in [\hat{x}_i, \hat{x}_{i+1}].$$

The next assertion is evident.

Lemma 7. *Suppose $t \in [\hat{x}_i, \hat{x}_{i+1}]$. If $u \in C^1[\hat{x}_i, \hat{x}_{i+1}]$ then the inequality $|u(t) - \tilde{u}(t)| \leq (\hat{x}_{i+1} - \hat{x}_i) \max_{\xi \in [\hat{x}_i, \hat{x}_{i+1}]} |u'(\xi)|$*

is fulfilled. If $u \in C^2[\hat{x}_i, \hat{x}_{i+1}]$ then the inequality

$$|u(t) - \tilde{u}(t)| \leq (\hat{x}_{i+1} - \hat{x}_i)^2 \max_{\xi \in [\hat{x}_i, \hat{x}_{i+1}]} |u''(\xi)|$$

holds.

Theorem 3. *Let $u \in C^1[a, b]$. Suppose the condition*

$$u'(t) \neq 0 \quad \forall t \in [a, b] \quad (31)$$

is true. If for $\eta > 0$ the grid (30) coincides with the grid

$\hat{X}(|u'|, \eta)$ *then*

1) *the quantity $K'_u(\eta) = K(|u'|, \eta)$ of knots satisfies relations*

$$\lim_{\varepsilon' \rightarrow 0} K(|u'|, \varepsilon') \varepsilon' = \int_a^b |u'| dt \leq K(|u'|, \varepsilon) + 1, \quad (32)$$

2) *the inequality*

$$|u(t) - \tilde{u}(t)| \leq \eta \quad \forall t \in [a, b] \quad (33)$$

is fulfilled.

Proof. Assuming $\hat{X} = \hat{X}(|u'|, \eta)$, we apply Lemma 3. As a result we have relation (32). Inequality (33) follows from Lemma 7 and formulas (3) – (4) for $f = |u'|$ and $\varepsilon = \eta$.

Theorem 4. *If $u \in C^2[a, b]$, the condition*

$$u''(t) \neq 0 \quad \forall t \in [a, b] \quad (34)$$

is fulfilled. If for $\eta > 0$ the grid (30) coincides with the grid $\hat{X}(|u''|, \eta)$ then

1) *the quantity $K''_u(\eta) = K(|u''|, \eta)$ of knots satisfies relations*

$$\begin{aligned} &\lim_{\varepsilon' \rightarrow 0} K(\sqrt{|u''|}, \varepsilon') \varepsilon' = \\ &= \int_a^b \sqrt{|u''|} dt \leq (K(\sqrt{|u''|}, \varepsilon) + 1), \end{aligned} \quad (35)$$

2) *the inequality*

$$|u(t) - \tilde{u}(t)| \leq \eta^2 \quad \forall t \in [a, b] \quad (36)$$

holds.

Proof. Analogously to the proof of the previous theorem we put $\hat{X} = \hat{X}(\sqrt{|u''|}, \eta)$. Implementing (6), we obtain relation (35). Inequality (36) follows from Lemma 7 and

relations (3) -- (4) for $f = \sqrt{|u''|}$, $\varepsilon = \eta^2$.

IV. ON QUANTITY OF KNOTS IN THE CASE OF AN EQUIDISTANT GRID

By the value $\eta > 0$ we construct an equidistant grid $\bar{X}(|u'|, \eta)$ with the step $h = (b - a)/N(|u'|, \eta)$, where $N(|u'|, \eta)$ is the quantity of knots for the mentioned grid.

Theorem 5. Consider $u \in C^1[a, b]$, $\eta > 0$. If the grid \bar{X} coincides with the grid $\bar{X}(|u'|, \eta)$ then

1) the number $N'_u(\eta) = N(|u'|, \eta)$ of knots satisfies to the relation

$$N(|u'|, \eta) = [(b - a) \max_{t \in [a, b]} |u'(t)| / \eta], \quad (37)$$

2) the inequality $|u(t) - \tilde{u}(t)| \leq \eta \quad \forall t \in [a, b]$ (38)

is true.

Proof. Setting $\bar{X} = \bar{X}(|u'|, \eta)$, we apply formula (19).

As a result we obtain relation (37). Inequality (38) follows from definition of the grid $\bar{X}(|u'|, \eta)$ (see Lemma 7 and formulas (19) -- (22), (25) for $f = |u'|$, $\varepsilon = \eta$).

Theorem 6. Suppose that $u \in C^2[a, b]$, $\eta > 0$. If the grid \bar{X} coincides with the grid $\bar{X}(|u''|, \eta)$ then

1) the number $N''_u(\eta) = N(\sqrt{|u''|}, \eta)$ of knots satisfies the relations

$$N(\sqrt{|u''|}, \eta) = [(b - a) \max_{t \in [a, b]} \sqrt{|u''(t)|} / \eta], \quad (39)$$

2) the inequality

$$|u(t) - \tilde{u}(t)| \leq \eta^2 \quad \forall t \in [a, b] \quad (40)$$

is right.

Proof. The proof of the theorem is similar to the proof of the previous one. Relations (39) -- (40) we obtain by formula (19), assuming $f = |u''|$ and $\varepsilon = \eta$.

The next assertion is true.

Theorem 7. Let a function $u \in C^1[a, b]$ with property

$$u'(t) \neq 0 \quad \forall t \in [a, b] \quad (41)$$

be fixed. If the deviations of the piece-wise linear interpolations

from the mentioned function u is equal to η then the ratio

$K'_u(\eta)/N'_u(\eta)$ has the next asymptotic behavior

$$\lim_{\eta \rightarrow 0} \frac{K'_u(\eta)}{N'_u(\eta)} = \frac{1}{b-a} \frac{\int_a^b |u'(t)| dt}{\max_{t \in [a, b]} |u'(t)|}. \quad (42)$$

Proof. Implementation of relations (32) -- (33) and

(37) -- (38) let us conclude that formula (42) is right under condition (41).

Theorem 8. Let a function $u \in C^2[a, b]$ have the property

$$u''(t) \neq 0. \quad (43)$$

Then

$$\lim_{\eta \rightarrow 0} \frac{K''_u(\eta)}{N''_u(\eta)} = \frac{1}{b-a} \frac{\int_a^b |u''(t)| dt}{\max_{t \in [a, b]} |u''(t)|}. \quad (44)$$

Proof. Usage of relations (35) -- (36) and (39) -- (40) allow us to obtain formula (44) under condition (43).

V. THE NUMBER OF OPERATIONS

First, we consider the adaptive grid construction algorithms

described in the proof of Lemma 1.

Theorem 9. If the conditions of Theorem 3 are fulfilled then

$$\begin{aligned} N(u') &= N_{add}(u') = N_{mul}(u') = K'_u(\eta) + M + 2, \\ N_{comp}(u') &= 2N_{mul}(u'). \end{aligned} \quad (45)$$

If the conditions of Theorem 4 are fulfilled then

$$\begin{aligned} N(u'') &= N_{add}(u'') = N_{mul}(u'') = K''_u(\eta) + M + 2, \\ N_{comp}(u'') &= 2N_{mul}(u''). \end{aligned} \quad (46)$$

The proof of formulas (45) -- (46) follows from Lemma 4, Theorem 3 and Theorem 4.

Consider the fine equidistant grid segment

$$[[a, b]] = \{a = \xi_0, \xi_1, \dots, \xi_{M+1} = b\},$$

where $\xi_i = i\hat{h}$, $\hat{h} = \frac{b-a}{M+1}$.

It is easy to see that the proof of Lemma 1 can be carried out so that the grid \bar{X} is a subset of the set $[[a, b]]$. In this case, the proof of Lemma 1 can be considered as an algorithm for constructing the grid \bar{X} . The number of operations in this algorithm is of the order of M .

Now we consider the construction of the approximation of this function in a situation in which the mentioned derivatives are not known.

We discuss the function

$$\Phi(\xi, x_0, x_1) = u(\xi) - u(x_0) - \frac{u(x_1) - u(x_0)}{x_1 - x_0} (\xi - x_0)$$

$$\forall \xi \in [x_0, x_1], \quad a \leq x_0 < x_1 \leq b, \quad \xi, x_0, x_1 \in \mathcal{E}. \quad (47)$$

The value $\Phi(\xi, x_0, x_1)$ can be called the deviation of the chord $L(x_0, x_1)$ from the function u at the point ξ . Consider the process of constructing the grid $\bar{X} = \bar{X}(u, \varepsilon)$, which consists of the fact that after finding the knot \tilde{x}_s , the knot \tilde{x}_{s+1} is searched using the two-point criterion:

1) $\Phi(\xi, x_s, x_{s+1}) \leq \varepsilon$ for $x_s < \xi < x_{s+1}$,

2) for a value $\delta > 0$ we have $\Phi(\xi, x_s, x_{s+1}) > \varepsilon$ for $x_{s+1} < \xi < x_{s+1} + \delta$.

This criterion is checked on the initial grid \mathcal{E} .

We have $\bar{X} \subset \mathcal{E}$. Consider the operation of searching for the difference

$$\Phi(\xi_i, \tilde{x}_s, \xi_j) = u(\xi_i) - u(\tilde{x}_s) - \frac{u(\xi_j) - u(\tilde{x}_s)}{\xi_j - \tilde{x}_s} (\xi_i - \tilde{x}_s). \quad (48)$$

The considered algorithm is iterative. It consists of the sequential selection of suitable knots of the initial grid. This algorithm will be described in more detail.

Algorithm (P):

0. Let $\tilde{x}_0 = a$.

1. Suppose the knots $\tilde{x}_0 < \tilde{x}_1 < \dots < \tilde{x}_s$ of the desired adaptive grid has already been found, and $\tilde{x}_s = \xi_{j_s}$. If $j_s + 1 \leq M$, then we change the parameter j in formula (48), sequentially taking $j = j_s + 2, j_s + 3, \dots, M + 1$ and checking every time the fulfillment of all inequalities $|\Phi(\xi_i, \tilde{x}_s, \xi_j)| \leq \varepsilon \quad i = j_s + 1, j_s + 2, \dots, j - 1. \quad (49)$

If all inequalities (49) are satisfied, and $j < M + 1$, then we add a unit to j and go back to checking inequalities (49), i.e. repeat point 1.

2. This process is interrupted in one of two cases:

a /. It turned out that $j = M + 1$. In that case, select the knot $\tilde{x}_{s+1} = \xi_{M+1}$ and put $K = s$. The construction of the adaptive grid \bar{X} is finished.

b /. For $j < M + 1$, at least one of the inequalities (49) is violated. In this case, we select the knot $\tilde{x}_{s+1} = \xi_{j-1}$. We

reassign variables by setting $s := s - 1$, $j_s := j - 1$ and go to step 1, i.e. make the next iteration cycle.

It is clear to see that the previous discussion proves the next assertion.

Theorem 10. *The number V of calculations of expression (48) satisfies the inequalities*

$$\frac{M+1}{K+1}(M-K) \leq 2V \leq (M-K+1)(M-K). \quad (50)$$

The left side of this inequality turns into equality, if the adaptive grid, which is found in accordance with the algorithm (P), turns out to be uniform. The right inequality (50) turns into equality if the mentioned grid can be formed removing one consecutive group of knots from the source grid.

VI. CONVEX FLOWS

Let ε be a positive value. Discuss the function $u(x)$ defined on the set \mathcal{E} .

Definition 1. We say that the function $u(x)$ is *weakly convex (up)* on the set \mathcal{E} if for any $a', b' \in \mathcal{E}$, $a' < b'$, inequality

$$u(\xi) \geq u(a') + \frac{u(b') - u(a')}{b' - a'}(\xi - a') \quad \forall \xi \in \mathcal{E}, \quad a' < \xi < b', \quad (51)$$

is right.

In this section, we assume that the arguments of the considered functions lie in the set \mathcal{E} . In particular, referring to representation (47), we assume that

$$x_0 \leq \xi \leq x_1, \quad x_0 < x_1, \quad x_0, \xi, x_1 \in \mathcal{E}.$$

It follows from formula (47) that the relations $\Phi(x_0, x_0, x_1) = \Phi(x_1, x_0, x_1) = 0$ are valid. If the function $u(x)$ is weakly convex (up), then

$$\Phi(\xi, x_0, x_1) \geq 0 \quad \text{for } \xi \in [x_0, x_1]. \quad (52)$$

To prove relation (52), it is sufficient to use the definition of the weak convexity (51), setting $a' = x_0$, $b' = x_1$.

Consider the supremum $F(x)$ of the function $\Phi(\xi, a, x)$ with respect to $\xi \in (a, x) \cap \mathcal{E}$, where $a < x$, $F(x) = \sup_{\xi \in (a, x) \cap \mathcal{E}} \Phi(\xi, a, x)$. It is clear to see that the next assertion holds.

Theorem 11. *If the function $u(x)$ is weakly convex (up) then $F(x') \leq F(x'')$ $\forall x', x'' \in (a, b) \cap \mathcal{E}$, $x' < x''$.*

Consider the algorithm for finding an adaptive grid, based on the idea of bisection. The algorithm described here is suitable for the weakly convex function. It is significantly more economical than the previous one.

We suppose that $b = \xi_{j_0}$, $F(\xi_{j_0}) \geq \varepsilon$.

Algorithm (Q):

0. We accomplish assignments $\tilde{x}_0 := \xi_0$, $s := 0$.

1. Suppose that the previous knot $\tilde{x}_s = \xi_i$ has been found i.e. $F(\xi_i) \leq \varepsilon$.

2. We do assignment $j := j_0$.

3. We calculate $k = \lfloor (i + j) / 2 \rfloor$.

If $F(\xi_k) \leq \varepsilon$ then we put $i := k$,

but if $F(\xi_k) > \varepsilon$ then we put $j := k$.

If $j - i > 1$ then we go to point 3, but if $j = i + 1$ then we put $\tilde{x}_{s+1} = \xi_i$. Thus the next knot of the adaptive grid has been found.

4. If $i < j_0 - 1$ then we put $s := s + 1$ and go to point

2, but if $i = j_0 - 1$ then the Algorithm (Q) has finished, because the adaptive grid has been obtained. A next fragment of the initial grid will begin with knot $a := b$.

Thus, the adaptive grid \mathcal{E} is determined by the sequential knot computation. Due to Theorem 10 (see also Corollary 1), the algorithm (Q) is effectively implemented. Amount \tilde{V} of computations of expression (48) has the order of $M \log_2 M$.

We give a two-sided estimate of the number \tilde{V} assuming that $M = 2^k$. We suppose that a relation from (48) is violated at each step of the algorithm (Q). In this case, the analyzed chords are constantly decreasing.

Ultimately we get $\tilde{x}_{s+1} = \xi_i$. The first step in implementing the algorithm (Q) is an analysis of the deviation of the chord $L(x_0, x_M)$ from the function u . Thus, the first step requires $2^k - 1$ calculations of expression (48). By convention, this deviation is greater than the value ε . The second step is an analysis of the deviation of the chord $L(x_0, x_{2^{k-1}})$ from the function u . It is evident that the second step requires $2^{k-1} - 1$ calculations of expression (48).

We suppose that this deviation is more than ε . Further we continue this way. Each time we suppose that the deviation of the considered chord from u is greater than ε . The final step is to consider the chord $L(\xi_0, \xi_2)$. It requires one calculation of expression (48).

General number \tilde{V}_0 of calculations for (48) presents with the formula

$$\tilde{V}_0 = \sum_{j=1}^k 2^j - 1 = 2^{k+1} - k - 2 \quad (53)$$

Consider the case of the deviation of the considered chords is not greater than the value ε . The final result is the assignment $\tilde{x}_{s+1} = \xi_{2^{k-1}-1}$. In this case, the length of the chords (starting from the third one) increases. As in the previous case, the first step is an analysis of the deviation of the chord $L(x_0, x_M)$ from the function u . It requires $2^{k-1} - 1$ calculations of expression (48).

By convention, this evasion is greater than the value ε . The second step is the analysis of the deviation of the chord $L(\xi_0, \xi_{2^{k-1}})$ from the function u . It requires $2^{k-1} - 1$ calculations of expression (48). We suppose that this deviation is not more than ε . The third step is the analysis of the deviation for the chord $L(\xi_0, \xi_{2^{k-1}+2^{k-2}})$ from the function u . It requires $2^{k-1} + 2^{k-2} - 1$ calculations of expression (48). We again suppose that gives the deviation not more than the value ε .

Continuing further this way, every time we suppose that the deviation of the considered chord from u is no more than the value ε . In the last step we discuss the deviation of the chord $L(\xi_0, \xi_q)$ from the function u . Here $q = 2^{k-1} + 2^{k-2} + \dots + 2^0$. The general number \tilde{V}_1 of calculations of formula (48) is presented by expression

$$\tilde{V}_1 = (2^k - 1) + (2^{k-1} - 1) + (2^{k-2} - 1) + \dots + (2^{k-1} + 2^{k-2} + \dots + 2^0 - 1) = k(2^k - 1). \quad (54)$$

By formulas (53) and (54) we see that the next assertion has been proved.

Theorem 12. *Suppose the Algorithm (Q) is implemented to*

a weakly convex flow $\{u(\xi_i)\}$, $i \in \{0,1,2, \dots, M\}$, where $M = 2^k$. Then the number \tilde{V} of calculations of formula (48) satisfies inequality

$$2^{k+1} - k - 2 \leq \tilde{V} \leq k(2^k - 1). \quad (55)$$

Inequality (53) is exact: the left and right parts can be reached for some flows.

Corollary 1. In the general case (that is, when M is not necessarily a power of two) under the conditions of Theorem 12, the estimate $2M - 1 - \lfloor \log_2 M \rfloor \leq \tilde{V} \leq (M - 1) \lfloor \log_2 2M \rfloor$ is right.

VII. DIGITAL EXPERIMENT

The purpose of the numerical experiment is to determine the effectiveness of the proposed adaptive algorithm for compression of the initial flow of numerical information. The initial flow $u(\xi_i)$ is generated by the function $u(t)$ using a uniform grid $\xi_i = i\hat{h}$, $i = 0, 1, \dots, M + 1$ on the segment $[a, b]$, $\hat{h} = \frac{b-a}{M+1}$. In this numerical experiment, we set $a = 0$, $b = \pi$, $M + 1 = 1000$. First, consider the standard compression $u(\xi_{2j})$, obtained by deleting nodes with odd numbers, $j=1, 2, \dots, N + 1$, $N + 1 = 500$. Let $\tilde{U}(t)$ be a piecewise linear interpolation (interpolation spline) constructed by points $(\xi_{2j}, u(\xi_{2j}))$, $j = 0, 1, 2, \dots, N + 1$. The deviation of this interpolation from the original flow is denoted ε , $\varepsilon = \max_{i=1, 2, 3, \dots, N-1} |u(\xi_{2i+1}) - \tilde{U}(\xi_{2i+1})|$. With the same uniform deviation ε we obtain the compressed flow of length $K+1$ using the proposed adaptive algorithm of compression for the initial flow. Here we use Theorem 4 for $\varepsilon = \eta^2$ with usual difference approximation of the second derivative. The ratio $(M + 1) / (K + 1)$ characterizes the degree of adaptive compression, the ratio $(N + 1) / (K + 1)$ characterizes the degree of efficiency of applying adaptive compression in comparison with standard double compression. In the digital experiment we discuss the generating functions $u(t) = \ln(t)$, $1/t$, $\ln(t + 1)$, $1/(t + 1)$ on the interval $[\hat{h}, \pi]$, and the function $u(t) = \cot(t)$ on the interval $[0.05, \pi - 0.05]$. In all cases we took $M + 1 = 1000$, $N + 1 = 500$. The results are shown in the Table 1. The first column contains the runtime of proposed adaptive algorithm for compression, second column contains generating function $u(t)$, the third column contains the deviation ε mentioned above, the fourth column gives the length $K + 1$ of the compressed flow after applying the proposed adaptive algorithm of compression to the original flow. The fifth and sixth columns contain characteristic of the degree of adaptive compression, characteristic of the degree of efficiency for applying adaptive compression in comparison with standard double compression.

Table 1. Computational results.

Time (sec)	$u(t)$	ε	$K+1$	$\frac{M+1}{K+1}$	$\frac{N+1}{K+1}$
0.28	$\ln(t)$	0.59	23	43.6	21.8
0.31	$1/t$	13.3	9	111	55.7
0.33	$\cot(t)$	0.062	61	16.4	8.21

0.33	$\ln(t+1)$	$5 \cdot 10^{-6}$	372	2.69	1.35
0.30	$1/(t+1)$	10^{-5}	289	3.46	1.73

Remarks to Table 1.

1. Instead of the second derivative, we used its approximation using the difference ratio.
2. For $\cot(t)$, instead of the interval $(0, \pi)$, the interval $(0.05, \pi - 0.05)$ was used.
3. The program is written in the Maple-17 system (see [34]). The calculations were carried out on an **HP 27-p251ur** monoblock. In the case of $M+1=10^6$ at $u(t)=\cot(t)$ took Time=70 sec. It turned out that $\varepsilon = 3 \cdot 10^{-5}$, $K + 1 = 3662$, $(M + 1) / (K + 1) = 273$, $(N + 1) / (K + 1) = 136$.
4. The calculation time for definition of the number ε in all cases was approximately 0.5 seconds

The results of a digital experiment show that

- 1) The proposed adaptive algorithm for compressing flows of numerical information in the case of rapidly changing flows, the compression ratio ranges from 16.4 to 111.
- 2) In the case of slowly changing flows, the compression ratio is much lower: it fluctuates between 2.69 and 3.46.
- 3) Behavior of the efficiency factor (relative to standard compression) similar to the behavior of the compression ratio (this ratio is half the compression ratio).
- 4) In the case of an increase in the length of the original flow, the coefficients of compression and efficiency increase significantly: for a flow of length $M+1=10^6$ for the generating function $u(t)=\cot(t)$ it turned out that $\varepsilon = 3 \cdot 10^{-5}$, $K + 1 = 3662$, compression ratios and efficiency equal to 273 and 136, respectively (see paragraph 3 in Remarks to Table 1).

VIII. CONCLUSION

In this paper we develop an adaptive compression algorithm with properties 1) the complexity of the algorithm is proportional to the length of the original flow, 2) by the piecewise linear interpolation of the compression result, it is possible to restore the original flow with an accuracy of $\varepsilon > 0$, 3) for rapidly changing smooth flows, the compression result is close to optimal. If an analog signal is discussed then the smoothness means the continuous differentiability (first or second order) of the function that generates the flow. If the initial flow is a discrete flow then the smooth means the boundedness of the appropriate difference ratios. Under discussion of the algorithm complexity, the number of computations of the original flow, the number of additive and multiplicative arithmetic operations, and the number of comparisons were separately taken into account. It is clear to see that the problem of the smallest numerical flow which can be used for piece-wise linear restoration $\tilde{u}(\xi_j)$ of the initial flow $u(\xi_j)$ with a predetermined precision $\varepsilon > 0$, has solution. This solution is required of $O(M^2)$ arithmetic operations (Section V contains the accurate estimates). Section

VI contains the improved algorithm, that gives $O(M \log_2 M)$ arithmetic operations in the case of convex flow. We propose the adaptive algorithm of compression, requiring $O(M)$ arithmetic operations in the case of smooth initial flow. The numerical experiment described in Section VII is consistent with the obtained theoretical results.

The approach used in this work is based on replacing the exact compression criterion with a weakened one. Instead of multiple calculations of the value $|\tilde{u}(\xi_j) - u(\xi_j)|$, we used one or another of its estimates with the property of monotonicity with an increase in the considered interval. By the mentioned weakening, instead of the optimal (best) solution to the problem of compressing the initial flow, we obtain a result of weaker quality. The advantage of this approach is that the number of arithmetic operations is sharply reduced. For a smooth initial flow of length M , instead of $O(M^2)$, we have $O(M)$ arithmetic operations. The considered approach can be applied in other cases: for compression using the generalized Haar approximation, splines of different orders, etc. The author hopes to take part in further research of this kind.

Acknowledgment

This work was partly supported by RFBR Grant 15-01-08847.

References

- [1] Marquardt D. Algorithm for least-squares estimation of nonlinear parameters. SIAM Journal on Applied Mathematics 1963; 11(2):431-441.
- [2] B.K. Swartz, and R.S. Varga, "Error bounds for spline and L-spline interpolation", J. Approximation 6, 1972, pp. 6-49.
- [3] L.L. Schumaker, "Constructive aspects of discrete polynomial spline functions", Approximation Theory, G.G. Lorentz, ed., Academic Press, New York, 1973.
- [4] Kazuo Toraichi, Kazuki Katagishi, Iwao Sekita & Ryoichi Mori. Computational complexity of spline interpolation// International Journal of Systems Science, Volume 18, 1987 - Issue 5, pages 945-954.
- [5] A.P. Pobegailo, "Local interpolation with weight functions for variable smoothness curve design", Comput.-Aided Des., 23 (8), 1991, pp. 579-582.
- [6] Friedman, J.H. Multivariate adaptive regression splines, The Annals of Statistics, 19, 1 (1991) 1-141.
- [7] Michael Unser, Akram Aldrobi, and Murray Eden. Fast B-Spline Transforms for Continuous Image Representation and Interpolation // IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.13, No. 3, March 1991, 277-285.
- [8] Shapiro L. Computer vision and image processing. Academic Press. 1992.
- [9] O. Roschel, "An interpolation subspline scheme related to B-spline techniques", Computer Graphics International, Proceedings, IEEE, 1997, pp. 131-136.
- [10] Stephane Mallat. A Wavelet Tour of Signal Processing. 2nd Edition. Academic Press. 1999.
- [11] Montemerlo M., Thrun S., Koller D., Wegbreit D. FastLAM: A factored solution to the simultaneous localization and mapping problem. Proc. Nat.Conf. Artif. Intell 2002: 593-598.
- [12] Forsyth D, Ponce J. Computer vision: A modern approach. Upper Saddle River, NJ: Prentice-Hall. 2003.
- [13] A. Wiltzsche. "Blending curves", J. Geom. Graph., 9 (1), 2005, pp. 67-75.
- [14] Durant-White H., Bailey T. Simultaneous localization and mapping (SIAM): Part I. The essential algorithms. IEEE Robot Automat Mag 2006; 13(2):99-110.
- [15] Durant-White H., Bailey T. Simultaneous localization and ... mapping (SIAM): Part II. State of art. IEEE Robot Automat Mag 2006; 13(3):108-117.
- [16] Larry Schumaker, *Spline Functions: Basic theory*, 3rd Edition, 2007.
- [17] D.J. Walton, D.S. Meek, "Hermite interpolation with circular precision", Comput.-Aided Des., 42 (9) 2010, pp. 749-758.
- [18] G.R. Liu, G.R. Zhang, "Edge-based Smoothed Point Interpolation Methods", International Journal of Computational Methods, 5(4), 2008, pp.621-646. DOI: 10.1142/S0219876208001510.
- [19] Kummerle R., Steder B., Dornhege C., Ruhnke M., Grisetti G., Stachniss C., Kleiner A. On measuring the accuracy of SLAM algorithms. Autonomous Robots 2009; 27(4): 387-407.
- [20] Z.Q. Zhang, G.R. Liu, "Upper and lower bounds for natural frequencies: A property of the smoothed finite element methods", International Journal for Numerical Methods in Engineering, 84, Issue: 2, 2010, pp.149-178. DOI: 10.1016/S0894-9166(13)60014-2.
- [21] Jörg Peters and Jianhua Fan. On the Complexity of Smooth Spline Surfaces from Quad Meshes // Computer Aided Geometric Design, Volume 27, Issue 1, January 2010, Pages 96-105
- [22] Newcombe R.A., Lovegrove S.J., Davison A.J. DTAM: dense tracking and mapping in real-time. IEEE Int. Conf. Computer Vision 2011: 2320-2327.
- [23] Eric Nyiri, Olivier Gibaru, Philippe Auquiart. Fast L1-Ck polynomial spline interpolation algorithm with shape-

preserving properties // Computer Aided Geometric Design, Elsevier, 2011, 28(1), pp.65-74.

[24] Levenberg K.A. Method for the solution of certain non-linear problems in least squares. Quarterly of Applied Mathematics 2012, 2(2):164-168.

[25] B. I. Kvasov, "Monotone and convex interpolation by weighted cubic splines", Comput. Math. Math. Phys., 53:10, 2013, pp. 1428-1439.

[26] I.G. Burova, T.O. Evdokimova, "On construction third order approximation using values of integrals", WSEAS Transactions on Mathematics, 13, 2014, pp. 676-683.

[27] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Hammerstein uniform cubic spline adaptive filters: Learning and convergence properties," //Signal Processing, vol. 100, pp. 112–123, 2014.

[28] I.G. Burova, Yu.K. Dem'yanovich. "On properties of decomposition operations for spline-wavelet representations", J. of Math. Science, 2015, 205, 2, Pp.205-221.

[29] Xiaogang Du, Jianwu Dang, Yangping Wang, Song Wang, and Tao Lei. A Parallel Nonrigid Registration Algorithm Based on B-Spline for Medical Images // Computational and Mathematical Methods in Medicine Volume, 2016, Article ID 7419307, 14 pages <http://dx.doi.org/10.1155/2016/7419307>

[30] Neira J, Reid I, Leonard JJ. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics 2016; 32(6): 1309-1332.

[31] I.G. Burova, O.V. Rodnikova, "Integro-differential polynomial and trigonometrical splines and quadrature formulae", WSEAS Transactions on Mathematics, 16, 2017, pp. 11-18.

[32] Roxas M., Oishi T. Real-time simultaneous 3D reconstruction and optical flow estimation. IEEE Winter Conference on Applications of Computer Vision (WACV) 2018: 885-893.

[33] Yu. K. Dem'yanovich, "Algorithms for Wavelet Decomposition of the Space of Hermite Type Splines", J. of Math. Science, 242, 1, 2019, pp.133-149.

[34] Maple 2017.0, Product Build IDs, Maple Build ID 1231047, Licensed to: Prof. Yuri Demyanovich, Serial Number: M4SUJR24AKMC7YDY, Permanent Licence.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en_US