Mathematics and Statistics 9(4): 411-420, 2021 http://www.hrpub.org
DOI: 10.13189/ms.2021.090401

Two-Stage Spline-Approximation with an Unknown Number of Elements in Applied Optimization Problem of a Special Kind

V. I. Struchenkov*, D. A. Karpov

Department of General Informatics, Institute of Cybernetics of the Russian Technological University (MIREA), Russia

Received March 10, 2021; Revised May 15, 2021; Accepted June 6, 2021

Cite This Paper in the following Citation Styles

(a): *[1] V. I. Struchenkov, D. A. Karpov , "Two-Stage Spline-Approximation with an Unknown Number of Elements in Applied Optimization Problem of a Special Kind," Mathematics and Statistics, Vol. 9, No. 4, pp. 411 - 420, 2021. DOI: 10.13189/ms.2021.090401.*

(b): *V. I. Struchenkov, D. A. Karpov (2021). Two-Stage Spline-Approximation with an Unknown Number of Elements in Applied Optimization Problem of a Special Kind. Mathematics and Statistics, 9(4), 411 - 420. DOI: 10.13189/ms.2021.090401.*

Abstract Being a continuation of the paper published in Mathematics and Statistics, vol. 7, No. 5, 2019, this article describes the algorithm for the first stage of spline-approximation with an unknown number of elements of the spline and constraints on its parameters. Such problems arise in the computer-aided design of road routes and other linear structures. In this article we consider the problem of a discrete sequence approximation of points on a plane by a spline consisting of line segments conjugated by circular arcs. This problem occurs when designing the longitudinal profile of new and reconstructed railways and highways. At the first stage, using a special dynamic programming algorithm, the number of elements of the spline and the approximate values of its parameters that satisfy all the constraints are determined. At the second stage, this result is used as an initial approximation for optimizing the spline parameters using a special nonlinear programming algorithm. The dynamic programming algorithm is practically the same as in the mentioned article published earlier, with significant simplifications due to the absence of clothoids when connecting straight lines and curves. The need for the second stage is due to the fact that when designing new roads, it is impossible to implement dynamic programming due to the need to take into account the relationship of spline elements in fills and in cuts, if fills will be constructed from soils of cuts. The nonlinear programming algorithm is based on constructing a basis in zero spaces of matrices of active constraints and adjusting this basis when changing the set of active constraints in an iterative process. This allows finding the direction of descent and solving the problem of excluding constraints from the active set without solving systems of linear equations in general or by solving linear systems of low dimension. As an objective function, instead of the traditionally used sum of squares of the deviations of the approximated points from the spline, the article proposes other functions, taking into account the specifics of a specific project task.

Keywords Route, Plan and Longitudinal Profile, Spline, Nonlinear Programming, Objective Function, Constraints, Basis

1. Introduction

In the theory of spline approximation and its numerous applications the problem of finding a spline with a known number of its elements and abscissas of their ends (spline nodes) is considered [1-5]. However, there are practically important problems in which both the boundaries and the number of elements of the required spline are unknown. In particular, such tasks include the design of linear structures routes (railways and highways, pipelines, canals, etc.).

A route is a 3D curve that is traditionally represented by two flat curves: a plan and a longitudinal profile.

The plan of the route is its projection onto the XOY plane, and the longitudinal profile is the graph of the function Z (s), where s is the length of the curve in the plan, measured from the given starting point. The longitudinal profile is obtained by unfolding a vertical cylindrical surface passing through the trace.

Regardless of the type of structure the design of a longitudinal profile can be considered as some spline building consisting of a given type elements. This spline should deviate minimally (in a given sense) from the original broken line, which is the profile of the earth in the design of new structures, and in the design of reconstruction, it is the profile of the route of the existing structure.

In the simplest case of using a first-order spline, the task is to transform the original broken line (ground profile) into another broken line that satisfies a number of constraints: on the slopes of elements and the difference in slopes of adjacent elements, on the minimum length of the element, on ordinates of individual points and zones (height constraints) [6,7,8]. Due to the smallness of the design slopes, the length of the element and the difference between the abscissas of its ends practically coincide; the difference in the slopes of adjacent elements is identified with the angle of rotation, and the slope is identified with the angle of the element with the OX axis. In this task, the number of elements of the required spline is unknown. This circumstance, as well as the presence of numerous constraints, significantly complicates the task.

It should also be noted that the objective function models are more complex in solving the design problem than the traditionally used minimization of the standard deviation.

The problem of finding the optimal spline in the form of a broken line with an unknown number of elements a constraints has been solved earlier when designing the longitudinal profile of new railway [6,7].

The problem was solved in two stages. At the first stage, the original profile was transformed into a broken line composed of short elements, observing all constraints, except for the restriction on the length of the element. This profile was named by the developers of the first design algorithms - "chain" [6].

At the second stage, the "chain" was transformed into a vertical alignment of railway, observing all constraints including the limitation on the length of the elements.

With regard to the longitudinal profile of roads when using second-degree parabolas and line segments as elements, the problem has been solved using a similar algorithm [9].

When designing high-speed railways, it became necessary to complicate the mathematical model, algorithm and previously developed design programs due to changes in technical requirements. The first order spline should be replaced with a spline consisting of straight lines and circular arcs, the number of which remains unknown.

A similar spline is used in the design of the longitudinal profile of large-diameter pipelines and roads as an alternative to the use of parabolas. A spline with circular arcs is also used when designing of routes plan of various linear structures [7].

The purpose of this article is to consider the above mentioned applied problems as problems of spline approximation from a unified theoretical standpoint, to outline the principal points of algorithms for their solution using nonlinear programming methods.

2. Problem Statement and Its Formalization at the First Stage

Let us consider the problem of designing a longitudinal profile with rectilinear elements that are conjugated by circular arcs.

If reconstruction is planned, then the profile of the existing structure is the initial one. If a new structure is being designed, the ground profile is the initial one. The "chain" longitudinal profile (dashed line in Fig.1) is used as a basis for defining the spline search area. To obtain the "chain", there are design programs [7]. Equality of the lengths of the elements of the "chain" is not required, but the abscissas of its nodes and the abscissas of the nodes of the original profile coincide.

So, we have a broken line Z (s), which needs to be transformed with minimal deviations into a spline z (s), consisting of line segments conjugating by circular arcs (Fig.1).

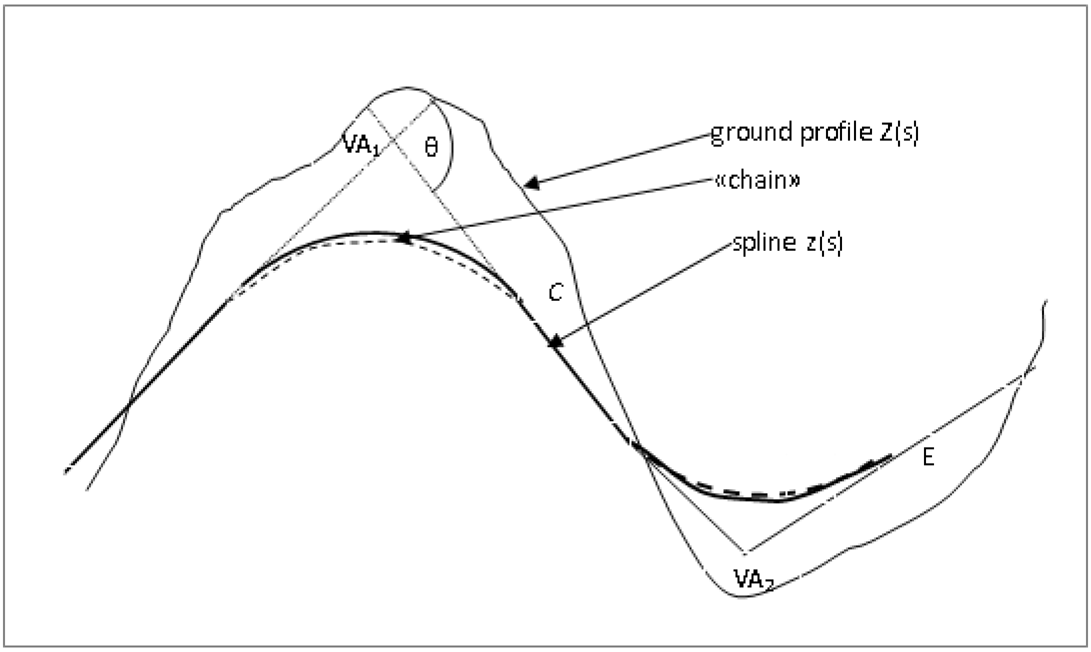

Figure 1. Spline with circular arcs

Constraints

1. on the slopes I_j of the straight-line elements of the spline: Imin ≤ I_j≤Imax. (j = 1,2…, N-1), where N is the number of vertices of the angles, hereinafter VA. Here Imin <0. In fact, this is a constraint on the first derivative of the function z(s).
2. on the radii of the inscribed convex and concave curves: Rmin≤ R_j≤Rmax. (j = 1,2, ..., N). Limit values of the radii of convex curves (at Rj <0) may differ from the corresponding values for concave curves.
3. on the lengths of circular arcs (BC, DE in Fig.1): Lcrj≥Lcrmin.
4. on the lengths of straight inserts between the curves (CD in Fig.1): Lsrj≥Lsrmin.

In addition, there may be constraints on the ordinates of some points (height constraints at the intersection of watercourses, other communications, etc.)

Objective function

Equal deviations in different directions from the baseline are not always equivalent in the problem under consideration. Therefore, the traditionally used sum minimization of the deviations squares at given points (including those with different weights) turns out to be unacceptable.

When designing new roads, the total amount of earthworks on fills and in cuts at this stage can be taken as the objective function. The cost of construction can be considered as an objective function if there is no relationship between the elements in the fills and in the cuts, which occurs when using soils from cuts for fill construction and requires taking into account the design line as a whole [7], (as in nonlinear programming). At the stage of transformation of the initial line ("chain" or existing profile) into a spline of the desired type, the deviations along the ordinates (working elevations) are not large (about 0.5 m [7]), which allows using simplified optimality criteria, since the goal of this stage is to determine the number of elements and their approximate location, that is, construct an initial approximation for the application of nonlinear programming.

When designing the reconstruction of the longitudinal profile of the road at this stage, it is advisable to use modeling functions, which take into account specific features of the problem.

So, when designing a longitudinal profile during reconstruction of a railway a smooth modeling function F (h) (a spline of the second order with a defect equal one) was successfully used, the graph of which is shown in Fig.2 [7]. Here h is the working elevations, i.e. the difference between the ordinates of the required and the original spline h (s) = z (s) -Z (s). The values of h_0, Δ and the parameters of the elements F (h) were determined based on the conditions of a specific problem.

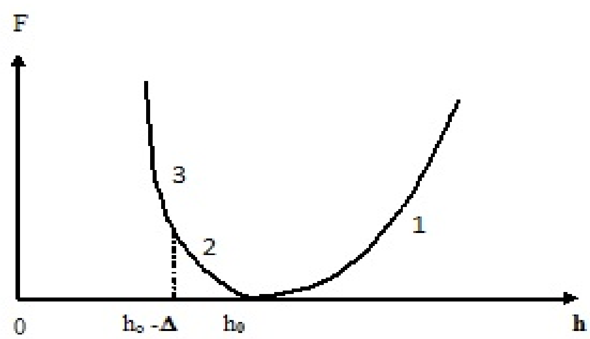

Figure 2. Modeling function

The values of F (h_i) were calculated at the nodes of the original spline, and the objective function was as follows:

$$\min\ \Phi(\boldsymbol{h}) = \sum_1^k v_i F(h_i), \qquad (1)$$

where k is the number of nodes of the original line, $\boldsymbol{h}$ ($h_1, h_2, \ldots, h_k$)- is a vector of unknowns, and the weight coefficients v_i are equal to the half-sums of the lengths of its adjacent elements. Similar modeling functions were used in the design of the longitudinal profile of the reconstructed roads with parabolic splines [7].

If the objective function is the volume of earthworks, then F (h_i) the cross-sectional area at the i-th point remains piecewise-quadratic and the objective function corresponds to the volume calculating as an integral of the area.

At the first stage, we take the coordinates of the VA and the radii of the inscribed circular curves as unknowns. In the implementation of dynamic programming, the key concept - "state of the system" is formalized as the point of the possible beginning of the next curve plus the angle of the tangent to the curve with the OX axis [8]. The radius of the vertical curve is determined by solving a one-dimensional optimization problem for each permissible position of the VA. As a result of the first stage, we obtain the coordinates of the VA and the radii of the inscribed curves.

3. Formalization of the Problem at Second Stage

Having determined the number of VAs and their abscissas, it is possible to formalize the problem of optimization of ordinates and radii as a nonlinear programming problem with a known admissible initial approximation.

Instead of fixing the abscissa of the spline nodes, we consider the abscissas of the VAs unchanged, that is, we consider the possibility of optimizing the position of the spline by moving the VAs along fixed verticals. Due to the smallness of the slopes (they do not exceed several tens of ppm), the lengths of the sides of each corner are considered equal to the difference between the abscissas VAs. Since the start and end points and directions are specified, the ordinates of the first and last VA cannot change. Therefore, we consider as variables only z_j (j = 1,2,…, n), i.e the ordinates of the VAs (their number n = N-2) and the radii of the inscribed curves R_j. The specified boundary conditions are taken into account by calculating the limit values for the slopes I_1 and I_n, and then the ordinates z_1 and z_n [7]. To obtain a nonlinear programming problem with an objective function Φ (h) (1) through these unknowns, it is necessary to express the working elevations at the nodes of the original polyline, i.e. the difference between the ordinates of the design spline and the original line (B′B″ in Fig.3), and all constraints.

When designing the longitudinal profile of new roads, the objective function corresponds to the minimum costs of earthworks and artificial structures. The corresponding models are the same as when using parabolic splines in the joint design of longitudinal and transverse profiles, taking into account the distribution of earth masses [7].

In the presence of such expressions, the calculation of the gradient of the objective function (1) is reduced to a simple recalculation of the derivatives [7], since the ordinates of the points (D, B in Fig.3) on the sides of the angle linearly depend on the ordinates C and A. Due to the smallness of the slopes, the turning angle is considered equal to the difference between adjacent slopes (ΔI_j in Fig.3).

This makes it possible, with sufficient accuracy for practice, to express the deviations of the points of the curve from the corresponding points of the straight lines (CC″, BB″ in Fig.3), that is, the corrections to the working elevations which were calculated along the sides of the angle of rotation.

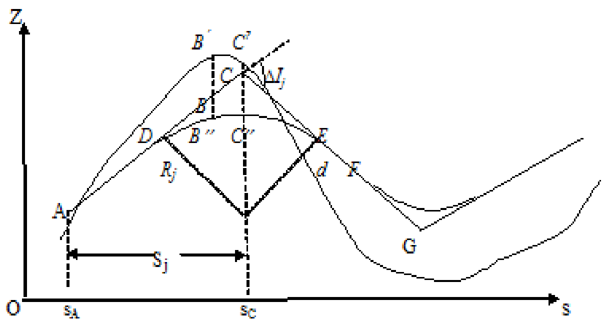

Figure 3. Calculation of derivatives in the presence of circular arcs

$CC'' = \delta_j \approx R_j \Delta I_j^2/8$; $BB'' = \delta_B \approx \delta_j - t_B\ \Delta I_j/2 + t_B^2/(2R_j)$, where $t_B = |s_C - s_B|$ – the difference between the abscissas of the VA and the points on the curve. $\Delta I_j = I_{j+1} - I_j$, and $I_j = (z_j - z_{j-1})/S_j$. Here z_j- unknown design elevations of the tops of the turning angles, and S_j are the differences of the abscissas, which, due to small slopes, do not significantly differ from the lengths of the sides of the angle.

$s_C - s_A = S_j \approx AC$ (Fig.3).

Instead of constraints on the difference in slopes, there are restrictions on the minimum lengths of the curves:

$R_j \Delta I_j \geq Lsr_j$ and on the minimum length of direct insertion EF, that is the sum CE+FG (Fig.3)

$$R_j \Delta I_j/2 + R_{j+1}\Delta I_{j+1}/2 + Lsrmin \leq S_{j+1}. \qquad (j=1,2, \ldots, n) \quad (2)$$

Here Lsrmin is the specified minimum length of the direct insertion, n is the number of VA.

At small ΔI_j , to change the length of the straight insert by 10 m, it is necessary to change the radius by 1000 m or more, which is unlikely when optimizing the spline obtained by the dynamic programming method. Therefore, the condition (2) can be simplified and the interconnection of variables related to adjacent VAs can be eliminated, using the spline obtained at the first stage as an initial approximation.

This can be done by following these steps:

1. Calculate all $T_j=R_j\Delta I_j/2$.
2. Calculate all direct inserts $v_j=S_j-(T_{j-1}+T_j)$. (j=2,…,n) and $w_j = v_j$- Lsrmin .
3. If v_j= Lsrmin, then T_{j-1} and T_j are fixed as maximum values of $R_{j-1}\Delta I_{j-1}/2$ and $R_j\Delta I_j/2$. We do not change the fixed values further.
4. Sequentially consider all direct inserts in ascending order, starting with the smallest v_k. The sum $T_{k-1}+T_k$ can be increased by $w_k = v_k$ – Lsrmin , without the risk of violating the constraint on direct insert on adjacent elements. If the maximum values $R_{k-1}\Delta I_{k-1}/2$ and $R_k\Delta I_k/2$ have not yet been fixed, then we take as their maximum values $T_{k-1}+ w_k/2$ and $T_k+ w_k/2$ respectively. w_{k-1} and w_{k+1} are reduced by $w_k/2$.

If T_{k-1} was fixed, then max $(R_k\Delta I_k/2)= T_k+ w_k$, and w_{k+1} decrease by w_k. If T_k is fixed then max $(R_{k-1}\Delta I_{k-1}/2)= T_{k-1}+ w_k$ and w_{k-1} decrease by w_k..

Go to step 3 and continue the process until there are no fixed maximum values $R_j\Delta I_j/2$. If necessary, the position of the start and end points of the profile is taken into account and the maximum values for $R_1\Delta I_1/2$ and $R_n\Delta I_n/2$ are corrected (downward).

Taking into account that $R_j\Delta I_j$ is the length of the j-th curve and Lcrmin its minimum value, denoting the calculated maximum values of $R_j\Delta I_j$ as $L_j$max, we obtain the system of two-sided inequalities

$$\text{Lcrmin} \leq R_j\Delta I_j \leq L_{j\,max} \quad (j=1,2,\ldots,n)$$

We transform this system of nonlinear inequalities into a linear system, passing from variable radii to curvatures $\sigma_j=1/R_j$. For constraints on $L_{j\,max}$ we have: $\Delta I_j \leq L_{j\,max}\sigma_j$ for $R_j>0$ and $L_{jmax}\ \sigma_j \leq \Delta I_j$ for $R_j<0$. For constraints on Lcrmin we obtain $L_\kappa\text{crmin}\ \sigma_j \leq \Delta I_j$ for $R_j>0$ and $\Delta I_j \leq L\text{crmin}\ \sigma_j$ for $R_j<0$. The signs of R_j are known, therefore, in general, for j = 1,2, ..., n we have linear constraints of four types:

1. on the slopes I_j : $I_{min} \leq I_j \leq I_{max}$;
2. on the curvatures σ_j: $\sigma_{jmin} \leq \sigma_j \leq \sigma_{jmax}$;
3. $\alpha_j\sigma_j \leq \Delta I_j \leq \beta_j\sigma_j$; If $R_j>0$ then $\beta_j= L_{j\,max}$ and $\alpha_j= L$crmin. For $R_j<0$, vice versa $\beta_j= L$crmin and $\alpha_j= L_{j\,max}$.
4. on ordinates of individual points.

4. The Main Items of the Nonlinear Programming Algorithm

We consider the main points of the iterative algorithm for solving the nonlinear programming problem in general terms:

Find min Φ (**x**), where **x** is the vector of unknowns, Φ (x) is the objective function, under linear constraints $\mathbf{Ax}\leq\mathbf{b}$.

1. Construction of an admissible initial approximation $\mathbf{x}^0$. Here the superscript is the iteration number k = 0;
2. Calculation of the anti-gradient **f**;
3. Formation of the matrix of active constraints Ak and construction descent vector p. This can be the projection of the anti-gradient to null- space of the matrix Ak, or any admissible direction for which (p, f)> 0;
4. If (p, p)> ε, then to the next item 5. Otherwise, there is the possibility check of excluding constraints from the active set. If there are no excluded constraints, the process is over. Otherwise, exclude one of them and with a new active set go to item 3. Modification of the active set is also possible for large components of the vector p, but in this case, it is necessary to provide measures against the zigzags discovered by Zoitendijk J.G [10].
5. Look for a step λ in the direction of descent at least steps to the border and to the minimum point. At this item, the one-dimensional problem of finding the minimum is solved [11].
6. Go to a new point $x^{k+1} = x^k + \lambda p$. Further to item 3, if the anti-gradient at the new point has already been calculated when searching for a step λ, otherwise – to item 2. In the general case, the algorithm leads to a neighborhood of a local minimum point. Therefore, after the dynamic programming, it is important to get a good initial approximation.

There are two key points: building the descent direction and excluding constraints from the active set. The problem can be solved by standard algorithms that require solving systems of linear equations at each iteration. The projection of the anti-gradient (we omit the iteration number) can be calculated using the Rosen formula:

$$p= (E- A^T(A\,A^T)^{-1}A)\ f \qquad (3)$$

To solve the problem of excluding constraints from the active set, it is necessary to calculate

$$u=(AA^T)^{-1}Af. \qquad (4)$$

The j-th constraint can be excluded if $\mathbf{u}_j<0$. The descent direction must be rebuilt, considering this constraint inactive.

5. An Alternative Approach

Instead, we will consider the possibility of a simple construction of a descent vector using a simple structure of constraints of all types, except for constraints on ordinates (type 4) [7]. To do this we must be able for any active set to construct a basis in the null - space of the matrix. This was implemented in the current spline optimization program in the form of a broken line without inscribed curves [7] under constraints of the form $\Delta I_j^{min} \leq \Delta I_j \leq \Delta I_j^{max}$.

If the basis matrix **C** is constructed, then projection p of the anti-gradient f onto the null - space of the matrix is calculated by the formula $\mathbf{p} = \mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1}\mathbf{C}^T\mathbf{f}$.

The reduced anti-gradient $\mathbf{p} = \mathbf{C}\mathbf{C}^T\mathbf{f}$ also can be taken as the descent vector, avoiding the solution of systems of linear equations [7, 11].

Constraints of type 3 additionally contain the variables σ_j, but the basis vectors can also be constructed for this system without cumbersome calculations.

Building a Basis

In our problem if some variable z_j is not included in any of the active constraints, then $\mathbf{p_j} = \mathbf{f}_j$. The presence of such free points makes it possible to split the profile into sections of independent construction of the basis vectors and the corresponding components of the descent vector **p**.

For example, for a system of active constraints of type 3 in the section with VAs from m + 1- to m + r-1- the variables are z_{m-1}, $z_m, \ldots, z_{m+r}$, z_{m+r} and σ_m , $\sigma_{m+1}, \ldots, \sigma_{m+r-1}$. Free variables are z_{m-2} and z_{m+r+1}.

Active constraints:

$$-\Delta I_m + \alpha_m \sigma_m \leq 0 \quad (\sigma_m<0) \tag{5}$$

$$\Delta I_{m+1} - \beta_{m+1}\sigma_{m+1} \leq 0 \quad (\sigma_{m+1}>0)$$

……………………….

$$-\Delta I_{m+r-1} + \alpha_{m+r-1}\sigma_{m+r-1} \leq 0 \quad (\sigma_{m+r+1}<0)$$

In variable ordinates, this system has the form:

$$-1/S_m z_{m-1} + (1/S_m + 1/S_{m+1}) z_m - 1/S_{m+1} z_{m+1} + \alpha_m\sigma_m \leq 0 \tag{6}$$

$$1/S_{m+1} z_m - (1/S_{m+1} + 1/S_{m+2}) z_{m+1} + 1/S_{m+2} z_{m+2} - \beta_{m+1}\sigma_{m+1} \leq 0$$

……………………………………………………………

$$-1/S_{m+r-1} z_{m+r-2} + (1/S_{m+r-1} + 1/S_{m+r}) z_{m+r-1} - 1/S_{m+r} z_{m+r} + \alpha_{m+r-1}\sigma_{m+r-1} \leq 0$$

The sought-for basis vectors must convert the inequalities of this system into equalities and be linearly independent. For example, the vector $\mathbf{c}^1 = (1\ 1\ \ldots\ 1\ 1\ 0\ 0\ 0)^T$ (r + 2 ones and r zeros) sets the shift of all VAs along the ordinate axis without changing the slopes and radii. Obviously, this does not change the curvature and the difference between adjacent slopes.

If we add equally to all the slopes (turn with the center at the m-1-th VA) and do not change the radii, then constraints of system (5) and the corresponding system (6) will remain active. Therefore, the vector

$$\mathbf{c}^2 = (0 \quad S_m \quad S_m + S_{m+1} \quad S_m + S_{m+1} + S_{m+2} \ldots \quad S_m + S_{m+1} + S_{m+2} + S_{m+r} \quad 0 \quad \ldots \quad 0)^T$$

can also be included in the required basis. Hereinafter, the superscript ^{T} means transposition. Another r basis vectors will be obtained by sequentially choosing m, m + 1,…, m + r-1 VAs as the centers of rotation, changing the slopes on the right by 1 and compensating for the change in the difference in slopes at the center of rotation by changing the corresponding curvature.

$$\mathbf{c}^3 = (0 \quad 0 \quad S_{m+1} \quad S_{m+1} + S_{m+2} \ \ldots \ S_{m+1} + S_{m+2} + \ldots + S_{m+r} \ \ldots \ 1/\alpha_m \quad 0 \ldots 0)^T$$

$$\mathbf{c}^4 = (0 \quad 0 \quad 0 \quad S_{m+2} \quad S_{m+2} + S_{m+3} \ldots S_{m+2} + S_{m+3} + \ldots + S_{m+r} \quad 0 \ldots 1/\beta_{m+1} \ldots \quad 0)^T$$

……………………………………………………………

$$\mathbf{c}^{r+2} = (0 \quad 0 \quad 0 \quad 0 \quad \ldots \quad 0 \quad \ldots . \quad S_{m+r} \quad 0 \ldots \quad 0 \ldots \quad 1/\alpha_{m+r-1})^T$$

We can set not the angle of rotation, but the increment (for example, 1) of the ordinate of the final VA and calculate the increment of the ordinates of the remaining VA by linear interpolation. The compensating component of the base vector will change accordingly. The linear independence of the obtained vectors follows from the method of their construction. We can continue building the basis using the turns if there are additionally active slope and curvature constraints on a similar site. There are many variants of the basis with several active constraints. A successful choice of one of them determines the complexity of computations when deciding whether to exclude constraints from the active set and the total amount of computations. The following approach seems appropriate:

1. The basis vector must contain the minimum number of nonzero components. For many combinations of active constraints, such vectors are easy to construct without computation.
2. When an additional active constraint appears, construct a new basis by transforming the existing one.
3. When excluding constraints from the active set, also modify the basis, but not build a new one.

Let us consider in general how to modify the basis matrix **C** when a new active constraint $(\mathbf{a}^j, \mathbf{x}) = b_j$ appears. Here $\mathbf{a}^j$ is the transposed j-th row vector of the matrix **A**, **x** is the vector of unknowns and b_j is the j-th component of the vector of free members of the system of constraints $Ax \leq b$. The number of basis vectors should decrease by 1.

Take a basis vector c_i for which $(a^j, \mathbf{c}^i) \neq 0$. If there is no such vector, then the new constraint does not change the null-space and the basis can be left unchanged.

All other basis vectors are transformed by the formula

$$\boldsymbol{c}^k_{new} = \boldsymbol{c}^k_{old} - \frac{(\boldsymbol{a}^j, \boldsymbol{c}^k_{old})}{(\boldsymbol{a}^j, \boldsymbol{c}^i)}\boldsymbol{c}^i \tag{7}$$

Here $\boldsymbol{c}^k_{old}$ and $\boldsymbol{c}^k_{new}$ are the original and transformed k-th basis vector. We exclude the vector $\boldsymbol{c}^i$ from the basis.

Formula (7) indeed gives a new basis, since for all $\mathbf{c}^k$ we have $(\mathbf{c}^k_{new}, \mathbf{a}^j) = 0$ and they are linearly independent, because any linear combination of them is a linear combination of the original basis vectors.

We will assume that the variables are ordered as follows: z_1 , z_2,…, z_n, σ_1, σ_2,…, σ_n.

If j curvature constraint is active, then $\mathbf{p}_{n+j}=0$. Accordingly $\mathbf{c}^i_{n+j}=0$, where $\mathbf{c}^i$ - any base vector. If a slope constraint is active, for example, $I_k=I_{max}$, i.e.

$z_k - z_{k-1}=S_k I_{max}$, then $\mathbf{p}_k = \mathbf{p}_{k-1}$ and $c^i_k = c^i_{k-1}$.

If only one constraint of type 3 is active, for example, $-\Delta I_m + \alpha_m\sigma_m \le 0$, i.e.

$-1/S_m z_{m-1}+(1/S_m +1/S_{m+1}) z_m- 1/S_{m+1} z_{m+1} + \alpha_m\sigma_m =0$, then it is necessary to construct three basis vectors, since the dimension of the corresponding null-space is equal to three. In accordance with the adopted approach, we get:

$$\mathbf{c^1}=(0...0\ 1\ 0…0\ c^1_{n+m-1}\ 0…0)^T,$$
where $c^1_{m-1}=1$, $c^1_{n+m-1}=1/(\alpha_m S_m)$;

$$\mathbf{c^2}=(0…\ 0\ 1\ 0…0\ c^2_{n+m}\ 0…0)^T,$$ where $c^2_m=1$, $c^2_{n+m}=-(1/S_m+1/S_{m+1})/\alpha_m$;

$$\mathbf{c^3}=(0…\ 01\ 0…0\ c^3_{n+m+1}\ 0…0)^T,$$ where $c^3_{m+1}=1$, $c^3_{n+m+1}=1/(\alpha_m S_{m+1})$;

There are only two nonzero components in each vector: one is equal to 1 and corresponds to a change in the ordinate of one VA, and the second compensates for the change in the slope difference in VA_m, the change in the slope difference in adjacent VA does not need to be compensated, since there is no active constraint in them.

Similarly, we will construct basis vectors in the presence of several active constraints of type 3, but it should be borne in mind that a change in one z_j can lead to a change in the difference in slopes in j-1, j and j + 1 VA. Accordingly, up to three compensating components appear in the base vector. In the compensating component of the basis vector, instead of α, there can be β with the same index, depending on which particular constraint is active and what is the sign of the curvature. Further, for definiteness, we use α, which corresponds to an active constraint for $\sigma > 0$ along the minimum length of the curve, and for $\sigma<0$, along the maximum length of the curve (along the straight insertion). Considering that at the extreme points of the section, the restrictions are inactive, and in other VAs an increase in z_j by 1 leads to changes ΔI_{j-1} by $1/S_j$, and ΔI_j by $-(1/S_j + 1/S_{j+1})$ and ΔI_{j+1} by $1/S_{j+1}$, which must be compensated for by changing respectively σ_{j-1}, σ_j and σ_{j+1}, we obtain the basis vectors:

$$\mathbf{c^1}=(0...0\ 1\ 0…0\ c^1_{n+m-1}\ 0…0)^T,$$
where $c^1_{m-1}=1$, $c^1_{n+m-1}=1/(\alpha_m S_m)$;

$$\mathbf{c^2}=(0…\ 0\ 1\ 0…0\ c^2_{n+m}\ c^2_{n+m+1}\ 0…0)^T$$ where $c^2_m=1$, $c^2_{n+m}=-(1/S_m+1/S_{m+1})/\alpha_m$, $c^2_{n+m+1}=1/(\alpha_{m+1}S_{m+1})$;

$$\mathbf{c^3}=(0…\ 0\ 1\ 0…0\ c^3_{n+m}\ c^3_{n+m+1}\ c^3_{n+m+2}\ 0…0)^T,$$ where $c^3_{m+1}=1$, $\mathbf{c}^3_{n+m}=1/(\alpha_m S_{m+1})$,

$c^3_{n+m+1}= -(1/S_{m+1}+1/S_{m+2})/\alpha_{m+1}$, $\mathbf{c}^3_{n+m+2}= 1/(\alpha_{m+2}S_{m+2})$;

………………………………………………………………

$$\mathbf{c}^{r+2}=(0\ \ …0\ \ …\ \ 010…\ \ 0c^{r+2}_{n+m+r-1}\ \ 0\ …\ 0)^T,$$ where $c^{r+2}_{m+r}=1$, $c^{r+2}_{n+m+r-1}=1/(\alpha_{m+r-1}S_{m+r})$;

If two sections of the considered type have one common VA, which corresponds to an inactive constraint of type 3, then for these sections the basis vectors are constructed as for a single whole. The basis vectors corresponding to the change in the final ordinate of the first section and the initial ordinate of the second section are replaced by one basis vector. For the example under consideration, this is a replacement $\mathbf{c}^{r+2}$ on

$$\mathbf{c}^{r+2}=(0\ \ …0\ \ …\ \ 010…\ 0c^{r+2}_{n+m+r-1}\ 0\ \mathbf{c}^{r+2}_{n+m+r+1}\ 0…0)^T,$$ where $c^{r+2}_{m+r}=1$,

$c^{r+2}_{n+m+r-1}=1/(\alpha_{m+r-1}S_{m+r})$; $c^{r+2}_{n+m+r+1}= 1/(\alpha_{m+r+1}S_{m+r+1})$;

The construction of the basis vectors is somewhat complicated when there are constraints of various types on the same site. However, it is possible to transform the basis according to the formula (7) with any "imposition" of constraints. In this case, only those basic types for which in the formula (7) $(\mathbf{a^j}, \mathbf{c^k})\neq 0$.

In our problem, when adding to the constraints of type 3, the constraints on the slope $\mathbf{a^j}$ contain two nonzero components, and when adding the constraint on the curvature, only one, therefore, many basis vectors do not change.

So, if in VA_i with an active constraint of type 3 the curvature also takes a limiting value, then in the additional constraint $a^j_{n+i}=1$, $a^j_k=0$ by $k\neq n+i$. Only the basis vectors corresponding to the change in the ordinates i-1, i and i + 1 of the VA give $(\mathbf{a}^j, \mathbf{c}^k)\neq 0$ in formula (7). Let's transform i-1 and i + 1 and exclude the i-th vector from the basis.

$(\mathbf{a}^j, \mathbf{c}^i)= -(1/S_i+1/S_{i+1})/\alpha_i$. For k=i-1 $(\mathbf{a^j}, \mathbf{c^k})=1/(\alpha_i S_i)$; and for k=i+1 $(\mathbf{a^j}, \mathbf{c^k})=1/(\alpha_i S_{i+1})$. Respectively

$$\mathbf{c}^{i-1}_{new}= \mathbf{c}^{i-1}_{old}+1/S_i/(1/S_i+1/S_{i+1})\ \mathbf{c}^i \text{ and}$$
$$\mathbf{c}^{i+1}_{new}= \mathbf{c}^{i+1}_{old}+1/S_{i+1}/(1/S_i+1/S_{i+1})\ \mathbf{c}^i.$$

Each of $\mathbf{c}^{i-1}_{new}$ and $\mathbf{c}^{i+1}_{new}$ has 5 non-zero components instead of 4. Naturally, their n + i-th component equal to 0. Let

$d_1 =1/S_i/(1/S_i+1/S_{i+1})$ and $d_2 =1/S_{i+1}/(1/S_i+1/S_{i+1})$. Then the nonzero components of the vector $\mathbf{c}^{i-1}_{new}$ are:

1, $-d_1$, $1/(S_{i-1}\alpha_{i-2})$, $-((1/S_{i-1}+1/S_i) - d_1/S_i)/\alpha_{i-1}$, $-d_1/(S_{i+1}\alpha_{i+1}))$ in places i-1, i, n+i-2, n+i-1 and n+i+1 respectively. Nonzero components of a vector $\mathbf{c}^{i+1}_{new}$ are:

$-d_2$, 1, $-d_2/(\alpha_{i-1}S_i)$, $(-(1/S_{i+1}+1/S_{i+2})-d_2/S_{i+1})/\alpha_{i+1}$, $1/(\alpha_{i+2}S_{i+2})$) in places i,i+1, n+i-1, n+i+1,n+i+2 respectively.

If on the considered section with active constraints of type 3 the limiting value is taken by the slope of some element I_i, then in formula (7) the vector $\mathbf{a^j}$ has two nonzero components $\mathbf{a}^j_{i-1}=-1/S_i$ and $\mathbf{a}^j_i=1/S_i$. It is necessary to convert only $\mathbf{c}^{i-1}$ and then exclude $\mathbf{c}^i$ (or vice versa). Exclude $\mathbf{c}^i$, which has 4 nonzero components:

1, $1/(S_i\alpha_{i-1})$, $-(1/S_i+1/S_{i+1})/\alpha_i$, $1/(S_{i+1}\alpha_{i+1})$) in places i, n+i-1, n+i, n+i+1 respectively. We get $(\mathbf{a^j}, \mathbf{c^i})=1/S_i$. Vector $\mathbf{c}^{i-1}$ also has 4 non-zero components:

1, $1/(S_{i-1}\alpha_{i-2})$, $-(1/S_{i-1}+1/S_i)/\alpha_{i-1}$, $1/(S_i\alpha_i)$ in places i-1, n+i-2, n+i-1, n+i respectively. $(\mathbf{a^j}, \mathbf{c}^{i-1})= -1/S_i$. Futher

(a^j, c^{i-1})/ (a^j, c^i)= -1 and by the formula (7) $c^{i-1}_{new}= c^{i-1}_{old}+c^i$.

To fulfill the conditions for fixed start and end points and directions, these conditions are converted into constraints of the form: $z_{1min} \leq z_1 \leq z_{1max}$ and $z_{nmin} \leq z_n \leq z_{nmax}$ [7].

If any of them becomes active, then, as follows from formula (7), the vector corresponding to the first and / or last VA is excluded from the basis, which was built without taking these constraints into account. The rest of the basis vectors is transformed according to the formula (7).

Difficulties arise in case of height restrictions at points on the inscribed curves of the form $z_{cr}^{min} \leq z_{cr} \leq z_{cr}^{max}$, since they nonlinearly depend on the ordinates of the VA (Fig.3). These constraints have to be taken into consideration using penalty functions [7]. As an alternative, it is possible to calculate the deviation δ of the corresponding point of the curve from the straight line connecting adjacent VAs (Fig.3) from the initial approximation, and replace z_{cr} by z_{sr} +δ for a concave curve δ> 0, and for a convex one δ <0. Here z_{sr} is the ordinate of a point on a straight line, depending linearly on the ordinates of adjacent VA.

The new linear constraint can be considered by transforming the basis by formula (7), built without taking it into_account. If after optimization the corresponding height limitation is inactive or δ has changed insignificantly, then the calculation is over. Otherwise, it is necessary to clarify δ and to continue the calculation. The choice of the way to take into account the height restrictions depends on the required accuracy of their implementation. Further, we assume that, as in the case of using parabolic splines [7], they are taken into consideration by penalty functions.

Modification of the Basis When Excluding Constraints from the Active Set

To exclude the constraint from the active set, we will include an additional vector in the existing basis.

When using the reduced anti-gradient $\mathbf{p=CC^Tf}$, the components of the vector $\mathbf{C^Tf}$ are the coordinates of the vector **p** in the basis **C**. Adding some vector **d** to the basis means calculating the reduced anti-gradient **p** by the formula $\mathbf{p}_{new} = \mathbf{p}_{old}$ + **(d,f)d,** where $\mathbf{p}_{old}$ is calculated using the old basis.

If $\mathbf{a}^j$ is the transposed row vector of the matrix of active constraints and ($\mathbf{a}^j$,**d**) =0 for all such rows except $\mathbf{a}^k$, then **d** is included in the basis if **(d,f)**> 0 and the sign of ($\mathbf{a}^k$, **d**) coincides with the sign of k- th inequality or **(d,f)**<0 and these signs are different. In other words, the vector **d** is included in the basis if $\mathbf{p}_{new}$ gives the next iteration point at which the k- th constraint is inactive and the value of the objective function has decreased.

If only one constraint of any kind is active on the site of independent construction of the descent direction, there is no need to construct an additional vector **d**. This constraint is eliminated if the corresponding components of the anti-gradient can be used as the components of the descent vector. Further, for various sets of active constraints, we will show how to construct a vector **d**, which must be checked for the possibility of being included in the basis for excluding a constraint from the active set.

If for any set of active constraints, an active constraint of type 3 in VA_j is checked and σ_j is not limiting, then $\mathbf{d}_{n+j}$=1 and the other components of the vector **d** are zeros.

At first, suppose that slope constraints are inactive and consider the possibility of excluding type 3 constraints with active curvature constraints. If we do not strive for the maximum number of zero components of the vector **d**, then we can use the rotation with the center at VA_j of the right (or left) part of the curve without compensating for the change in ΔI_j.

The possibility of obtaining zero components in the vector **d** depends on the presence and location of VA, in which the limitation 3 is inactive or the curvature is not limiting. In the presence of such "free" VA on the left (VA_i closest to VA_j) and (VA_k, respectively, on the right) in the zero vector **d**, we set $\mathbf{d}_j$ = 1. The components **d** (increments of the ordinates of the curve) with numbers i <r <j and j <r <k are determined using linear interpolation. We ignore the changes in ΔI_i if the constraint of type 3 in VA_i is inactive, or we compensate it by specifying the corresponding $\mathbf{d}_{n+i}$. We do the same with VA_k.

If there are "free" VA only to the right or only to the left of VA_j, for example, with numbers i <k <j, then in the zero vector **d** we set $\mathbf{d}_k$ = 1 and determine the components with numbers i <m <k and k <m <j by linear interpolation with the subsequent calculation of $\mathbf{d}_{n+i}$ and $\mathbf{d}_{n+k}$ to compensate for the resulting difference in slopes, if the type 3 constraints in these VA are active.

If the exclusion of the constraint on σ_j is checked, then compensation for the change in ΔI_j is additionally required.

Now suppose that in addition to active type 3 and curvature constraints, there are active slope constraints. The solution to the problem of excluding the constraint of type 3 in the VA_j with the limiting σ_j depends on the numbers of the limiting slopes, the presence and position of "free" VAs.

The solution of this issue is not affected by the limiting slopes that are outside the considered interpolation sections. Otherwise, if the limiting slopes are located on both sides of the investigated VA_j, then at least up to one of them there must be a “free” point k. For example, j <k <m, where m is the limit slope number. In this case, the rotation with the center at VA_j is performed up to point k. For the zero vector **d**, we put $\mathbf{d}_k$ = 1. On the interval j <i <k, the **d** components are calculated by linear

interpolation and $\mathbf{d}_i = 1$ for i> k. If there are “free” points for i> m, then we can use them to go to zero components.

At the same time, compensation for the change in the difference in slopes in those VA is performed, where it is necessary, as it was in the absence of limiting slopes (see above). Rotation of the left side of the curve can be used similarly.

A special case is when constraints of type 3 and curvature are active between the limiting slopes I_m and I_{m+r} in all VA_j (m <j <m + r), i.e. no "free" points. In this case, it is impossible to change ΔI_j while keeping activity of all other constraints. It is advisable to start by excluding curvature and slope constraints. To exclude the slope constraint I_j from the active set, there must be $\mathbf{d}_{j-1} \neq \mathbf{d}_j$, the other components of the vector **d** must not change the set of active constraints. To the left or to the right of VA_j, in the general case, there should be a "free" VA_k. Vector **d** is constructed using a rotation centered at VA_k and calculating the **d** components as previously indicated, including calculating $\mathbf{d}_{n+k}$ to compensate for the change in ΔI_k if necessary. In the presence of "free" points, further transformation of the rotation is possible, both in the presence of other limiting slopes, and to preserve the zero components of the vector **d**.

6. Discussion

Various options for organizing the iterative optimization process are possible. So, the issue of excluding a constraint from the active set can be considered at any iteration. It is not necessary to supplement the basis. For example, if the exclusion of the constraint leads to the appearance of two separate sections for plotting the direction of descent instead of one, then it makes sense to correct some basis vectors and consider sections separately. It is characteristic that the proposed above method for finding a constraint, which can be excluded from the active set, in contrast to the use of formulas (3,4), allows, when some conditions are met, to simultaneously exclude several constraints. It was experimentally found that if we take measures against zigzags, it reduces the counting time. However, this is not guaranteed for all tasks.

Various algorithms can be implemented, in particular, work sequentially with separate sections of the independent construction of the descent direction, including the modification of the active set, and combining such sections when active constraints occur at their ends. The alternative is to change all coordinates at each iteration.

7. Conclusions

Two-stage spline approximation can be used in intelligent systems for designing linear structures routes and can give a significant economic effect, primarily due to the optimization of the spline parameters in comparison with the interactive design [12-15]. It was established in the 70-80s of the last century [6] during the practical application of programs for designing the longitudinal profile of railways using splines in the form of broken lines and highways using parabolic splines [9] on a computer of that time. This can be expected from the use of programs that design splines with circular arcs on modern computers. The approach outlined in the article can find application in solving problems not related to the design of linear structures routes. In particular, formula (7) allows not only transforming the basis under any linear constraints, but also constructing it starting from the unit matrix.

In the design of linear structures routes plan the problems of optimizing splines with variable abscissas of the boundaries of elements arise, which, moreover, are not single-valued functions. A separate article will be devoted to their solution.

REFERENCES

[1] J. H. Alberg, E. N. Nilson, J. L. Walsh “the Theory of Splines and Their Application”. New York, Academic Press, 1967, pp. 1-316.

[2] V.A. Vasilenko. “Optimization of energy functional for variational splines “, Bull of the Novosibirsk Computing Center, Num. Anal. Novosibirsk: NCC Publisher, no 7, pp. 101-107, 1996.

[3] H. N. Zayniddinov, Madhusudan Singh. “Polynomial Splines for Digital Systems”, LAMBERT Academic publishing, Germany, 2019, pp.1-208.

[4] Dhananjay Singh, Madhusudan Singh, Zaynidinov Hakimjon. “Signal Processing Applications Using Multidimensional Polynomial Splines”, Springer Briefs in Applied Sciences and Technology, Springer, Singapore, 2019, 1-70pp. https://doi.org/10.1007/978-981-13-2239-6_5

[5] Hakimjon Zaynidinov, Jonibek Juraev, Umidjon Juraev, “Digital Image Processing with Two Dimensional Haar Wavelets”, International Journal of Advanced Trends in Computer Science and Engineering,2020,vol.9,no 3, pp. 2729-2734. Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse38932020.pdf https://doi.org/10.30534/ijatcse/2020/38932020

[6] “The use of mathematical optimization techniques and computer engineering at the longitudinal profile of the railways”. / Ed. by B.K. Malyavsky. Moscow, Transport, 1977. pp.1-169. (Proceedings of the All-Union Research Institute of Transport Construction. Iss. 101).

[7] V. I Struchenkov. “Methods to optimize the routes in the CAD systems of linear structures.” Moscow, Solon – Press, 2014, pp.1-271.

[8] D.A.Karpov, V. I. Struchenkov. “Dynamic Programming as a Method of Spline Approximation in the CAD Systems of Linear Constructions”. Russian Technological Journal, 2019, vol 7, no 3, pp. 78-88, DOI:10.32362/2500-316X-2019-7-3-77-88

[9] Struchenkov V.I., Karikh Yu.S., Schwartz P.S. “Mathematical methods of optimization in the computer-aided design of roads”. Highways, no. 12, 1980, pp 21-29

[10] Zoitendijk J.G. “Methods of Possible Directions”. Moscow.: IL, 1963,pp.1-214

[11] Philip F. Gill,Walter Murray, Margaret H. Wright, “Practical optimization”, Emerald Group ane, Bingley Bg16, UK,1982, pp. 1-389.

[12] ”Bentley ProjectWise 365”. https://www.bentley.com/ru/products/product-line/project-delivery- software/projectwise-365 (vizited: Feb. 2,.2021).

[13] ” CARD/1.” URL: https://www.card-1.com/en/home/ (vizited: Feb.20,2021).

[14] “Autodesk AutoCAD”. URL: https://www.autodesk.ru/products/autocad/overview?term=1-YEAR&support=null (vizited: Feb.26,.2021).

[15] “CREDO software products and technologies”: official website. Credo-Dialog. URL: https://credo-dialogue.ru/ (vizited: Feb.14,.2021).