УДК 519.65

О РАВНОМЕРНОМ ВОССТАНОВЛЕНИИ ФУНКЦИЙ, ИМЕЮЩИХ НЕ БОЛЕЕ ДВУХ ТОЧЕК ПЕРЕГИБА

Нечипоренко Н. А., к. ф.-м. н., доцент, Коротунова Е. В., к. т. н., доцент

Запорожский национальный технический университет,
ул. Жуковского, 64, г. Запорожье, 69063, Украина

kafedra_pm@zntu.edu.ua

Рассматривается задача восстановления непрерывной функции, заданной своими приближенными значениями в узлах произвольной фиксированной сетки и имеющей в области определения не более двух точек перегиба. В качестве восстанавливающей принимается функция, построенная на основе метода квазирешений. Приводимые алгоритмы восстановления являются оптимальными по порядку точности на соответствующих классах функций.

Ключевые слова: функция одной переменной, восстановление, изогеометрические свойства, оптимальность по точности.

ПРО РІВНОМІРНЕ ВІДНОВЛЕННЯ ФУНКЦІЙ, ЩО МАЮТЬ НЕ БІЛЬШЕ ДВОХ ТОЧОК ПЕРЕГИНУ

Нечипоренко Н. О., к. ф.-м. н., доцент, Коротунова О. В., к. т. н., доцент

Запорізький національний технічний університет,
вул. Жуковського, 64, м. Запоріжжя, 69063, Україна

kafedra_pm@zntu.edu.ua

Розглядається задача відновлення неперервної функції, яка задана своїми наближеними значеннями у вузлах довільної фіксованої сітки і має в області визначення не більше двох точок перегину. Як відновлююча приймається функція, побудована на основі методу квазірозв’язків. Наведені алгоритми відновлення є оптимальними за порядком точності на відповідних класах функцій.

Ключові слова: функція однієї змінної, відновлення, ізогеометричні властивості, оптимальність за точністю.

ON UNIFORM RESTORATION OF FUNCTIONS WHICH HAVE NO MORE THAN TWO POINTS OF INFLECTION

Nechyporenko N. A., Ph.D., Associate Professor, Korotunova O. V., Ph.D., Associate Professor

Zaporizhzhya National Technical University,
Zhukovsky street, 64, Zaporizhzhya, 69063, Ukraine

kafedra_pm@zntu.edu.ua

The problem of tabular data approximation is one of the major problems arising in the processing of the experimental results. In many cases, the experimenter has additional information about the geometric properties of the restored function, which are advisable to take into account and save. Many works are dedicated to the constructing interpolation isogeometric splines problem. However, the application area of these splines is limited to tables containing the exact values of the interpolated function. Generally, in practice there are approximate values that do not match the available priori information about the geometric properties of the functions. Therefore, the problem of restoration functions algorithms constructed by theirs approximate gridded data and a priori information about theirs geometric properties is actual.

This work considers the problem of restoring a continuous function from a class of functions defined by theirs approximate values in arbitrary fixed grid nodes. As a set of functions, were considered the following sets: concave and convex monotone; having a domain not more than two points of inflection and monotone function with the same property. Restoring function is based on the quasi-solutions method. The step-by-step algorithms for restoring the function of each of the classes of functions are provided. The numerical results show that these algorithms can achieve high accuracy of the recovery. If the uncertainty of the input data at the grid is known and the corresponding class of functions is limited, it is possible to construct upper and lower bounds of the corresponding class of functions. Then, as is known, for optimum accuracy of approximation in the class will be a Chebyshev center of a class equal to half the sum of the upper and lower boundaries of the class. However, in this case the desired geometric properties of functions will not be performed, which is unacceptable. Quoted in the work function approximation algorithms retain its geometric properties and are optimal in order of accuracy, with the constant orders not exceeding 2.

Key words: function of one variable, restoration, isogeometric properties, optimality on accuracy.

ВВЕДЕНИЕ

Одной из основных задач, возникающих при обработке результатов экспериментов, является задача восстановления функций, заданных таблицами своих приближенных значений. При этом часто возникает необходимость в сохранении восстанавливающей функцией определенных свойств – монотонности, выпуклости и т.д. Эти свойства могут быть получены из априорных представлений о поведении тех или иных физических, экономических процессов или явлений, описываемых искомыми функциями. Использование качественной априорной информации о функции позволяет строить регуляризирующие алгоритмы для широкого класса некорректных задач путем минимизации невязки на классе функций, удовлетворяющих априорным ограничениям [1]. Для реализации этих алгоритмов требуются значительные вычислительные затраты. В работах [2-4] рассмотрены так называемые изогеометрические сплайны, которые сохраняют геометрию приближаемой функции, но при очень существенном условии: табличные значения должны соответствовать геометрическим свойствам функции. К сожалению, в эксперименте регистрируются, как правило, «зашумленные» значения функции, которые чаще всего не соответствуют имеющейся априорной информации. Для сглаживающих сплайнов в работе [5] предложена методика учета априорной информации, заданной в виде ограничений на значения функции и ее производных в узлах сетки. Недостатком данной методики является то, что она гарантирует заданное поведение только в узлах сетки, хотя в задачах обработки экспериментальных данных необходимо выполнять априорные ограничения на всем заданном интервале.

Эмпирический анализ большого числа природных, технико-экономических и социокультурных процессов показал, что динамика их роста, развития подчиняется логистическому закону, то есть описывается неубывающими функциями, имеющими одну точку перегиба. Поэтому построение по экспериментальным данным оптимальных алгоритмов восстановления выпуклых функций, а также функций, имеющих одну или две точки перегиба, является актуальной задачей.

ПОСТАНОВКА ЗАДАЧИ

Пусть функция $f(x)$, принадлежащая некоторому классу F определенных на $[a,b]$ функций, задана своими приближенными значениями f_j, $j=\overline{1,N}$ в узлах x_j произвольной фиксированной сетки $\Delta:\{a=x_1<x_2<...<x_N=b\}$. Требуется восстановить эту функцию. В качестве восстанавливающей принимается функция $S(x)$, которая принадлежит классу F и построена на основе метода квазирешений, то есть удовлетворяет условию:

$$\delta(S)=\inf_{\phi\in F}\delta(\varphi), \tag{1}$$

$$\delta(\varphi)=\max_{1\le i\le N}\left|f_i-\varphi(x_i)\right|,$$

где инфимум берется по всем функциям $\varphi(x)$ из класса F.

Пусть $F_{N,\varepsilon}$ – класс функций $f(x)\in F$ и удовлетворяет условию $\left|f(x_i)-f_i\right|\le\varepsilon$, $i=\overline{1,N}$. Так как функция $S(x)$, являющаяся решением задачи (1), принадлежит классу $F_{N,\varepsilon}$, то при условии ограниченности этого класса, восстановление функцией $S(x)$ является оптимальным по порядку точности в классе $F_{N,\varepsilon}$ с константой порядка, не превосходящей 2.

В качестве F рассмотрим следующие классы функций:

– $V_i[a,b]$, $i=1,2,3$ – класс непрерывных функций $f(x)$, $x\in[a,b]$, минимальное число интервалов выпуклости которых на $[a,b]$ не превосходит i;

– $W_i[a,b]$, $i=1,2,3$ – класс непрерывных функций $f(x)\in V_i[a,b]$ и монотонных на $[a,b]$.

Отметим, что имеющиеся экспериментальные данные f_j, $j=\overline{1,N}$ о функции $f(x)$, как правило, не соответствуют имеющейся априорной информации о геометрических свойствах функции $f(x)$.

ВОССТАНОВЛЕНИЕ ВЫПУКЛЫХ И МОНОТОННЫХ ВЫПУКЛЫХ ФУНКЦИЙ

Рассмотрим задачу восстановления непрерывных выпуклых функций: найдем функцию $S(x)$, которая удовлетворяет условию (1) при $F\equiv V_1[a,b]$. Для определенности будем считать, что функции класса $V_1[a,b]$ выпуклые вниз.

Рассмотрим следующий пошаговый

Алгоритм А.

Шаг 1. Положить $s=1$, $k_0=1$.

Шаг 2. Вычислить $\dfrac{f_{k_s}-f_{k_{s-1}}}{x_{k_s}-x_{k_{s-1}}}=\min\limits_{k_{s-1}<i\le N}\dfrac{f_i-f_{k_{s-1}}}{x_i-x_{k_{s-1}}}$ и зафиксировать k_s.

Шаг 3. Найти $\delta_s=1/2\max\limits_{k_{s-1}\le i\le k_s}\left(f_i-y_s(x_i)\right)$, где $y_s(x)=f_{k_{s-1}}+\dfrac{f_{k_s}-f_{k_{s-1}}}{x_{k_s}-x_{k_{s-1}}}\left(x-x_{k_{s-1}}\right)$.

Шаг 4. Проверить: $k_s=N$? Если да, то перейти к шагу 6, иначе к шагу 5.

Шаг 5. Увеличить s на единицу и перейти к шагу 2.

Шаг 6. Найти $\delta(s)=\max\limits_{1\le i\le s}\delta_i$.

Шаг 7. Положить $z_i(x)=y_i(x)+\delta_i$, $i=\overline{1,s}$.

Шаг 8. Положить $S(x)=\max\limits_{1\le i\le s}z_i(x)$, $x\in[a,b]$ и закончить вычисления.

Доказано, что алгоритм A дает решение задачи (1) при $F\equiv V_1[a,b]$. При этом решение $S(x)$ обладает следующим локальным свойством: $\left|f_i-S(x_i)\right|\le\min\limits_{\varphi\in V_1[a,b]}\max\limits_{k\in K_j}\left|f_k-\varphi(x_k)\right|$, $i\in\left\{l\,|\,x_l\in\left[\overline{z}_j,\overline{z}_{j+1}\right],1\le l\le N\right\}$, где $\overline{z}_p$, $p=1,2,...,m-1$, $m\le s$ – точки разрыва производной функции $S(x)$, $\overline{z}_0=x_1$, $\overline{z}_m=x_N$, $K_j=\left\{l\,|\,x_l\in\left[\overline{z}_{j-1},\overline{z}_{j+2}\right],1\le l\le N\right\}$, $j=\overline{1,m-2}$.

Рассмотрим теперь задачу восстановления непрерывных выпуклых монотонных функций, то есть задачу (1) при $F\equiv W_1[a,b]$. Для определенности будем считать, что функции класса $W_1[a,b]$ $W_1[a,b]$ являются непрерывными неубывающими выпуклыми вниз функциями.

Рассмотрим следующий пошаговый

Алгоритм А$_1$.

Шаг 1. Найти $f_n=\min\limits_{1\le i\le N}f_i$ и зафиксировать n.

Шаг 2. Проверить: $n=1$? Если да, то перейти к шагу 3, иначе к шагу 4.

Шаг 3. Выполнить шаги 1-8 алгоритма A и закончить вычисления.

Шаг 4. Найти $\delta_1 = 1/2 \max\limits_{1 \le i \le n} \left(f_i - f_n \right)$ и положить $y_1\left(x\right) = f_n$.

Шаг 5. Проверить: $n = N$? Если да, то положить $S\left(x\right) = y_1\left(x\right) + \delta_1$, $x \in \left[a,b\right]$ и закончить вычисления.

Шаг 6. Положить $s = 2$, $k_1 = n$.

Шаг 7. Выполнить шаги 2-8 алгоритма A и закончить вычисления.

Доказано, что алгоритм A_1 дает решение задачи (1) при $F \equiv W_1\left[a,b\right]$, при этом решение $S\left(x\right)$ обладает таким же локальным свойством, что и в случае $F \equiv V_1\left[a,b\right]$.

ВОССТАНОВЛЕНИЕ ФУНКЦИЙ, ИМЕЮЩИХ НЕ БОЛЕЕ ДВУХ ТОЧЕК ПЕРЕГИБА

Будем говорить, что точка x^* является точкой перегиба функции $f\left(x\right)$, если существует число $\delta > 0$ такое, что во всех точках $x \in \left(x^* - \delta, x^*\right)$ функция $f\left(x\right)$ выпуклая вверх (вниз), а во всех точках $x \in \left(x^*, x^* + \delta\right)$ функция $f\left(x\right)$ выпуклая вниз (вверх). В соответствии с этим определением минимальное число точек перегиба функций класса $V_2\left[a,b\right]$ не превосходит единицы, а функций класса $V_3\left[a,b\right]$ – не превосходит двойки.

Построим решение задачи (1) при $F \equiv V_2\left[a,b\right]$. Для определенности будем считать, что функции класса $V_2\left[a,b\right]$ выпуклые вниз на $\left[a, x^*\right)$ и выпуклые вверх на $\left(x^*, b\right]$, где x^* – произвольная точка $\left[a,b\right]$.

Рассмотрим следующий пошаговый

Алгоритм B.

Шаг 1. Положить $s = 1$, $k_0 = 1$, $p = 1$, $n_0 = N$.

Шаг 2. Вычислить $\dfrac{f_{k_s} - f_{k_{s-1}}}{x_{k_s} - x_{k_{s-1}}} = \min\limits_{k_{s-1} < i \le n_{p-1}} \dfrac{f_i - f_{k_{s-1}}}{x_i - x_{k_{s-1}}}$ и зафиксировать k_s.

Шаг 3. Найти $\delta_s^l = 1/2 \max\limits_{k_{s-1} \le i \le k_s} \left(f_i - y_s^l\left(x_i\right)\right)$, где $y_s^l\left(x\right) = f_{k_{s-1}} + \dfrac{f_{k_s} - f_{k_{s-1}}}{x_{k_s} - x_{k_{s-1}}}\left(x - x_{k_{s-1}}\right)$.

Шаг 4. Вычислить $\dfrac{f_{n_p} - f_{n_{p-1}}}{x_{n_p} - x_{n_{p-1}}} = \min\limits_{k_{s-1} < i \le n_{p-1}} \dfrac{f_i - f_{n_{p-1}}}{x_i - x_{n_{p-1}}}$ и зафиксировать n_p.

Шаг 5. Найти $\delta_p^r = 1/2 \max\limits_{n_p \le i \le n_{p-1}} \left(y_p^r\left(x_i\right) - f_i \right)$, где $y_p^r\left(x\right) = f_{n_{p-1}} + \dfrac{f_{n_p} - f_{n_{p-1}}}{x_{n_p} - x_{n_{p-1}}}\left(x - x_{n_{p-1}}\right)$.

Шаг 6. Проверить: $\delta_s^l < \delta_p^r$? Если да, то перейти к шагу 7, иначе к шагу 9.

Шаг 7. Вычислить $z_s^l\left(x\right) = y_s^l\left(x\right) + \delta_s^l$, $S_s^l\left(x\right) = \max\limits_{1 \le i \le s} z_i^l\left(x\right)$.

Шаг 8. Положить $s = s + 1$, $j = 1$ и перейти к шагу 11.

Шаг 9. Вычислить $z_p^r\left(x\right) = y_p^r\left(x\right) - \delta_p^r$, $S_p^r\left(x\right) = \min\limits_{1 \le i \le p} z_i^r\left(x\right)$.

Шаг 10. Положить $p = p + 1$, $j = 0$.

Шаг 11. Перейти к шагу 13, если $s=1$ или $p=1$.

Шаг 12. Проверить: $S_{s-1}^{l}\left(x_{k_{s-1}}\right) > S_{p-1}^{r}\left(x_{k_{s-1}}\right)$ и $S_{s-1}^{l}\left(x_{n_{p-1}}\right) > S_{p-1}^{r}\left(x_{n_{p-1}}\right)$? Если да, то перейти к шагу 15.

Шаг 13. Проверить: $k_{s-1}+1 < n_{p-1}$? Если да, то перейти к шагу 2.

Шаг 14. Положить $S(x)=\begin{cases} S_{s-1}^{l}(x), & \text{если} \quad a \le x \le x_{k_{s-1}}; \\ g(x), & \text{если} \quad x_{k_{s-1}} \le x \le x_{n_{p-1}}; \\ S_{p-1}^{r}(x), & \text{если} \quad x_{n_{p-1}} \le x \le b, \end{cases}$ где

$g(x)=g\left(x,x_{k_{s-1}},S_{s-1}^{l}\left(x_{k_{s-1}}\right),x_{n_{p-1}},S_{p-1}^{r}\left(x_{n_{p-1}}\right)\right)$, если $s>1$ и $p>1$;

$g(x)=g\left(x,x_1,f_1,x_{n_{p-1}},S_{p-1}^{r}\left(x_{n_{p-1}}\right)\right)$, если $s=1$;

$g(x)=g\left(x,x_{k_{s-1}},S_{s-1}^{l}\left(x_{k_{s-1}}\right),x_N,f_N\right)$, если $p=1$;

$g(x)=g\left(x,u_1,\upsilon_1,u_2,\upsilon_2\right)=\dfrac{\upsilon_2-\upsilon_1}{u_2-u_1}\left(x-u_1\right)+\upsilon_1$ и закончить вычисления.

Шаг 15. Проверить: $s=2$ или $p=2$? Если да, то положить $w=x_1 sign(p-2)+x_N sign(s-2)$ и перейти к шагу 21.

Шаг 16. Проверить: $j=1$? Если да, то перейти к шагу 18.

Шаг 17. Определить w такое, что $S_{p-2}^{r}(w)=z_{p-1}^{r}(w)$ и перейти к шагу 19.

Шаг 18. Определить w такое, что $S_{s-2}^{l}(w)=z_{s-1}^{l}(w)$.

Шаг 19. Проверить: $S_{s-1}^{l}(w)\le S_{p-1}^{r}(w)$? Если да, то перейти к шагу 21.

Шаг 20. Положить $s=s-1$, если $j=1$ и $p=p-1$, если $j=0$.

Шаг 21. Положить $S(x)=\begin{cases} S_{s-1}^{l}(x), & \text{если} \quad x_1 \le x \le x^*; \\ S_{p-1}^{r}(x), & \text{если} \quad x^* \le x \le x_N, \end{cases}$, где

$x^*=\min\left\{x \middle| S_{s-1}^{l}(x)=S_{p-1}^{r}(x),\ \min\left(w,x_{k_{s-2}}\right)\le x\le x_{n_{p-1}}\right\}$, если $j=1$;

$x^*=\max\left\{x \middle| S_{s-1}^{l}(x)=S_{p-1}^{r}(x),\ x_{k_{s-1}}\le x\le \max\left(w,x_{n_{p-2}}\right)\right\}$, если $j=0$.

Доказано, что функция $S(x)$, построенная в результате выполнения алгоритма B, является решением задачи (1) при $F\equiv V_2[a,b]$.

Построим теперь решение задачи (1) при $F\equiv W_2[a,b]$. Для определенности будем считать, что функции класса $W_2[a,b]$ неубывающие, выпуклые вниз на $\left[a,x^*\right)$ и выпуклые вверх на $\left(x^*,b\right]$, где x^* – произвольная точка $[a,b]$.

Рассмотрим следующий пошаговый

Алгоритм B₁.

Шаг 1. Положить $s=1$, $k_0=1$, $p=1$, $n_0=N$.

Шаг 2. Проверить: $s>1$? Если да, то перейти к шагу 6.

Шаг 3. Найти $f_{k_s}=\min\limits_{1\le i\le n_{p-1}} f_i$ и зафиксировать k_s.

Шаг 4. Проверить: $k_s=1$? Если да, то перейти к шагу 6.

Шаг 5. Найти $\delta_1^l=1/2\max\limits_{1\le i\le k_s}\left(f_i-f_{k_s}\right)$, положить $y_1^l(x)=f_{k_s}$ и перейти к шагу 8.

Шаг 6. Вычислить $\dfrac{f_{k_s}-f_{k_{s-1}}}{x_{k_s}-x_{k_{s-1}}}=\min\limits_{k_{s-1}<i\le n_{p-1}}\dfrac{f_i-f_{k_{s-1}}}{x_i-x_{k_{s-1}}}$ и зафиксировать k_s.

Шаг 7. Найти $\delta_s^l=1/2\max\limits_{k_{s-1}\le j\le k_s}\left(f_j-y_s^l\left(x_j\right)\right)$, где $y_s^l(x)=f_{k_{s-1}}+\dfrac{f_{k_s}-f_{k_{s-1}}}{x_{k_s}-x_{k_{s-1}}}\left(x-x_{k_{s-1}}\right)$.

Шаг 8. Проверить: $p>1$? Если да, то перейти к шагу 12.

Шаг 9. Найти $f_{n_p}=\max\limits_{k_{s-1}\le i\le N} f_i$ и зафиксировать n_p.

Шаг 10. Проверить: $n_p=N$? Если да, то перейти к шагу 12.

Шаг 11. Найти $\delta_1^r=1/2\max\limits_{n_p\le i\le N_s}\left(f_{n_p}-f_i\right)$, положить $y_1^r(x)=f_{n_p}$ и перейти к шагу 14.

Шаг 12. Вычислить $\dfrac{f_{n_p}-f_{n_{p-1}}}{x_{n_p}-x_{n_{p-1}}}=\min\limits_{k_{s-1}\le i<n_{p-1}}\dfrac{f_i-f_{n_{p-1}}}{x_i-x_{n_{p-1}}}$ и зафиксировать n_p.

Шаг 13. Найти $\delta_p^r=1/2\max\limits_{n_p\le i\le n_{p-1}}\left(y_p^r\left(x_i\right)-f_i\right)$, где $y_p^r(x)=f_{n_{p-1}}+\dfrac{f_{n_p}-f_{n_{p-1}}}{x_{n_p}-x_{n_{p-1}}}\left(x-x_{n_{p-1}}\right)$.

Шаг 14. Выполнить шаги 6-21 алгоритма B и закончить вычисления.

Алгоритм B_1 отличается от алгоритма B только определением величин δ_1^l, $y_1^l(x)$ и δ_1^r, $y_1^r(x)$. Это вызвано необходимостью соблюдения условий монотонности функций $S_{s-1}^l(x)$, $S_{p-1}^r(x)$. Доказано, что функция $S(x)$, построенная в результате выполнения алгоритма B_1, является решением задачи (1) при $F\equiv W_2[a,b]$.

Основываясь на алгоритмах B и B_1, можно привести алгоритмы, позволяющие найти решение задачи (1) на множествах непрерывных, неубывающих или невозрастающих непрерывных функций при различном расположении участков выпуклости и вогнутости на $[a,b]$. Так, например, решение $\tilde{S}(x)$ задачи (1) на множестве непрерывных функций, выпуклых вверх на $\left[a,x^*\right)$ и выпуклых вниз на $\left(x^*,b\right]$, где x^* – произвольная точка отрезка $[a,b]$, можно получить в результате выполнения следующего алгоритма $\tilde{B}$:

Шаг 1. Положить $f_i'=-f_i$, $i=\overline{1,N}$.

Шаг 2. Построить функцию $S(x)$, $x\in[a,b]$, являющуюся результатом применения алгоритма B к входным данным $x=\{x_1,x_2,...,x_N\}$, $f=\{f_1',f_2',...,f_N'\}$.

Шаг 3. Положить $\tilde{S}(x)\equiv -S(x)$, $x\in[a,b]$.

Аналогично может быть построен алгоритм $\tilde{B}_1$ решения задачи (1) на множестве неубывающих непрерывных функций, выпуклых вверх на $\left[a,x^*\right)$, выпуклых вниз на $\left(x^*,b\right]$, где x^* – произвольная точка отрезка $\left[a,b\right]$.

В заключение заметим, что основываясь на алгоритмах восстановления выпуклых функций и функций, имеющих не более одной точки перегиба, можно построить алгоритмы восстановления функций, имеющих не более двух точек перегиба, а именно построить алгоритмы решения задачи (1) при $F\equiv V_3\left[a,b\right]$ и $F\equiv W_3\left[a,b\right]$. Для определенности будем считать, что функции класса $V_3\left[a,b\right]$ выпуклые вниз, если $x\in\left[a,x^*\right)$ или $x\in\left(x^{**},b\right]$ и выпуклые вверх, если $x\in\left(x^*,x^{**}\right)$, где x^* и x^{**} – произвольные точки отрезка $\left[a,b\right]$, $a\le x^*\le x^{**}\le b$.

Сохраняя обозначения, принятые в алгоритмах A и B, рассмотрим следующий

Алгоритм С.

Шаг 1. Применить алгоритм A к входным данным $x=\left\{x_1,x_2,...,x_N\right\}$, $f=\left\{f_1,f_2,...,f_N\right\}$ и вычислить k_i, $y_i\left(x\right)$, δ_i, $i=\overline{1,s}$.

Шаг 2. Вычислить $\delta^0=\max\limits_{1\le i\le s}\delta_i$ и зафиксировать m такое, что $\delta^0=\delta_m$.

Шаг 3. Проверить: $\delta^0=0$? Если да, то положить $\alpha=\beta=N$, в противном случае положить $\alpha=k_{m-1}$, $\beta=k_m$.

Шаг 4. Определить $f_\nu-y_m\left(x_\nu\right)=\max\limits_{\alpha\le i\le\beta}\left(f_i-y_m\left(x_i\right)\right)$ и зафиксировать x_ν.

Шаг 5. Положить $S_1\left(x\right)\equiv S\left(x\right)$, $x\in\left[x_1,x_\nu\right]$, где $S\left(x\right)$ является результатом применения алгоритма B к входным данным $x=\left\{x_1,x_2,...,x_\nu\right\}$, $f=\left\{f_1,f_2,...,f_\nu\right\}$.

Шаг 6. Положить $S_2\left(x\right)\equiv\tilde{S}\left(x\right)$, $x\in\left[x_\nu,x_N\right]$, где $\tilde{S}\left(x\right)$ является результатом применения алгоритма $\tilde{B}$ к входным данным $x=\left\{x_\nu,x_{\nu+1},...,x_N\right\}$, $f=\left\{f_\nu,f_{\nu+1},...,f_N\right\}$.

Шаг 7. Положить $S\left(x\right)=\begin{cases}S_1\left(x\right), & x_1\le x\le\tilde{x},\\ S_2\left(x\right), & \tilde{x}<x\le x_N,\end{cases}$ где $\tilde{x}\in\left[x_\alpha,x_\beta\right]$ и является корнем уравнения $S_1\left(x\right)=S_2\left(x\right)$, если $\delta^0>0$ и $\tilde{x}=x_N$, если $\delta^0=0$. Закончить вычисления.

Доказано, что функция $S\left(x\right)$, построенная в результате выполнения алгоритма C, является решением задачи (1) при $F\equiv V_3\left[a,b\right]$.

Рассмотрим теперь задачу восстановления монотонных функций класса $V_3\left[a,b\right]$: построим решение задачи (1) при $F\equiv W_3\left[a,b\right]$. Алгоритм C_1 решения этой задачи идентичен алгоритму C решения задачи (1) при $F\equiv V_3\left[a,b\right]$. Отличие состоит только в том, что на шагах 1, 5 и 6 алгоритма C алгоритмы A, B, $\tilde{B}$ необходимо заменить на соответствующие алгоритмы A_1, B_1, $\tilde{B}_1$.

РЕЗУЛЬТАТЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ

Опишем некоторые из проведенных численных экспериментов восстановления функций. Пусть функция $f\left(x\right)$ принадлежит одному из рассматриваемых классов функций $V_i\left[a,b\right]$

или $W_i[a,b]$, $i=1,2$. Входная информация для решения задачи восстановления формировалась следующим образом. На отрезке $[a,b]$ выбиралась равномерная с шагом h сетка x_i, $i=\overline{1,N}$. На этой сетке вычислялись значения функции $f(x_i)$, $i=\overline{1,N}$, на которые затем накладывался некоторый вектор ошибок δ_i, $i=\overline{1,N}$ и в качестве исходных данных использовались $y_i = f(x_i)+\delta_i$, $i=\overline{1,N}$. При этом величины δ_i, $i=\overline{1,N}$ являются равномерно распределенными случайными величинами, удовлетворяющими условию $\max\limits_{1\le i\le N}\delta_i = \frac{1}{10}\max\limits_{1\le i\le N}|f(x_i)|$.

В качестве модельных примеров рассматривались следующие: $f_1(x)=\frac{1}{x}$, $x\in[1;10]$; $f_2(x)=e^x$, $x\in[0,1;1]$; $f_3(x)=\ln x$, $x\in[1,1;2]$; $f_4(x)=\sin x$, $x\in[0;1,35]$; $f_5(x)=\sin x$, $x\in[-1;1]$; $f_6(x)=\frac{\sin x}{x}$, $x\in[-2\pi;-0,1]$; $f_7(x)=ctgx$, $x\in[0,5;2]$; $f_8(x)=e^{-x^2}$, $x\in[-1;0]$. С помощью программной реализации приведенных в работе алгоритмов для каждой из функций $f_k(x)$, $k=\overline{1,8}$ решалась задача восстановления по информации x_i, y_i, $i=\overline{1,N}$ для 30 различных наборов δ_i, $i=\overline{1,N}$, которые формировались с помощью датчика случайных чисел. Обозначим через $S_k^{(j)}(x)$ восстанавливающую функцию, соответствующую j-му набору величин δ_i, $i=\overline{1,N}$. Точность восстановления будем оценивать величиной $E\left(S_k^{(j)}\right)$, определяемой по формуле

$$E\left(S_k^{(j)}\right)=\max_{x\in\bigcup\limits_{i=1}^{N-1}\Delta_i}\left|f_k(x)-S_k^{(j)}(x)\right|,\quad k=\overline{1,8},$$

где Δ_i – равномерная сетка на отрезке $[x_i,x_{i+1}]$ с шагом $(x_{i+1}-x_i)/10$. Для каждой серии из 30 задач восстановления функции $f_k(x)$, $k=\overline{1,8}$ в процентах определено количество задач $n(f_k)$, для которых величина $E\left(S_k^{(j)}\right)$, $j=\overline{1,30}$ принадлежит одному из интервалов $[0;0,25\delta]$, $(0,25\delta;0,5\delta]$, $(0,5\delta;0,75\delta]$, $(0,75\delta;\delta]$, $(\delta;1,25\delta]$, где $\delta=\max\limits_{1\le i\le N}\delta_i$. Результаты расчетов при $N=10$ приведены в таблице 1.

Таблица 1

$n(f_k)$, $k=\overline{1,8}$	Область изменения $E\left(S_k^{(j)}\right)$, $j=\overline{1,30}$				
	$[0;0,25\delta]$	$(0,25\delta;0,5\delta]$	$(0,5\delta;0,75\delta]$	$(0,75\delta;\delta]$	$(\delta;1,25\delta]$
$n(f_1)$	3,3	13	50,6	13	20
$n(f_2)$	3,3	30	53,3	10	3,3
$n(f_3)$	3,3	63,3	23,3	10	-
$n(f_4)$	3,3	20	60	16,7	-
$n(f_5)$	-	30	50	20	-
$n(f_6)$	-	20	60	20	-
$n(f_7)$	-	23,3	53,3	23,3	-
$n(f_8)$	-	33,3	46,6	20	-

ВЫВОДЫ

В статье приведены алгоритмы восстановления таблично заданной функции $f(x)\in F$, которые могут быть использованы при обработке экспериментальных данных. Эти алгоритмы позволяют не только сохранить изогеометрические свойства восстанавливаемой функции $f(x)$, но и, как показывают результаты численных экспериментов, достичь достаточно высокой точности восстановления. В качестве восстанавливающей принимается функция $S(x)$, построенная на основе метода квазирешений. Если задана точность ε входных данных $f(x_i)$, $i=\overline{1,N}$ и соответствующий класс функций $F_{N,\varepsilon}$ ограничен, то приведенные алгоритмы являются оптимальными по порядку точности с константой, не превосходящей 2. Отметим также, что используя значения x_i, $S(x_i)$, $i=\overline{1,N}$, можно построить гладкую восстанавливаемую функцию с требуемыми геометрическими свойствами.

ЛИТЕРАТУРА

1. Тихонов А. Н. Регуляризующие алгоритмы и априорная информация / А. Н. Тихонов, А. В. Гончарский, В. В. Степанов, А. Г. Ягола. – М. : Наука, 1983. – 200 с.
2. Квасов Б. И. Методы изогеометрической аппроксимации сплайнами / Б. И. Квасов. – М. : Физматлит, 2006. – 360 с.
3. Гребенников А. И. Метод сплайнов и решение некорректных задач теории приближений / А. И. Гребенников. – М. : Изд-во МГУ, 1983. – 206 с.
4. Мирошниченко В. Л. Достаточные условия монотонности и выпуклости для интерполяционных кубических сплайнов класса С / В. Л. Мирошниченко // Приближение сплайнами. – Новосибирск, 1990. – Вып. 137 : Вычислительные системы. – С. 31-40.
5. Воскобойников Ю. Е. Дескриптивные сглаживающие сплайны и алгоритмы их построения / Ю. Е. Воскобойников // Моделирование в механике. Сб. научных трудов. Новосибирск, Институт теоретической и прикладной механики СО РАН. – 1991. – Т. 5, № 5. – С. 30-37.

REFERENCES

1. Tikhonov, A.N., Goncharsky, A.V., Stepanov, V.V. and Yagola, A.G. (1983), *Regulyariziruyushchiye algoritmy i apriornaya informatsiya* [Regularizing algorithms and a priori information], Nauka, Moscow.
2. Kvasov, B.I. (2006), *Metody izogeometricheskoy approksimatsii splaynami* [Isogeometric priori splines approximation methods], FIZMATLIT, Moscow.
3. Grebennikov, A.I. (1983), *Metod splaynov i resheniye nekorrektnykh zadach teorii priblizheniy* [Splines method and some ill-posed problems in approximation theory solution], MGU, Moscow.
4. Miroshnichenko, V.L. (1990), “Sufficient conditions of monotonicity and convexity for interpolating cubic splines C”, *Vychislitel'nyye sistemy*, vol. 137, Novosibirsk, pp. 31-40.
5. Voskoboynikov, Y.E. (1991), “Descriptive smoothing splines and algorithms for their construction”, *Modelirovaniye v mekhanike*, vol. 5, no. 5, Novosibirsk, pp. 30-37.