DOI 10.37882/2223–2966.2023.03.03

ПРИМЕНЕНИЕ АЛГОРИТМОВ КОМПРЕССИИ С ЦЕЛЬЮ СОКРАЩЕНИЯ ОБЪЕМА ВЫЧИСЛИТЕЛЬНЫХ ДАННЫХ

APPLICATION OF COMPRESSION ALGORITHMS TO REDUCE THE VOLUME OF COMPUTATIONAL DATA

A. Brusova

Summary. In this paper, we analyzed algorithms for compressing point arrays in order to find the most efficient one. The research method is the use of an algorithm (and its simplified version) based on the criteria for excluding points using the height formula, and an algorithm for finding the maximum and minimum on the segment. Obtained visual results of their action. Based on the algorithms, a program has been developed to facilitate the receipt of conclusions.

Keywords: compression, compression algorithm, array decimation, compression based on maximum and minimum.

Брусова Анна Александровна
Аспирант, МИРЭА — Российский технологический университет
SupernaturalAnn@yandex.ru

Аннотация. В данной работе выполнен анализ алгоритмов сжатия массивов точек с целью нахождения наиболее эффективного. Методом исследования является применение алгоритма (и его упрошенной версии), основанного на критерии исключения точек при помощи формулы высоты , и алгоритма поиска максимума и минимума на отрезке. Получены наглядные результаты их действия. На основе алгоритмов разработана программа, для облегчения получения выводов.

Ключевые слова: сжатие, компрессия, алгоритм сжатия, прореживание массива, сжатие на основе максимума и минимума.

Введение

В современном мире есть потребность хранить, передавать и использовать для различных целей большие объемы данных. Одним из путей решения проблем передачи и т.д., является компрессия данных. Компрессия данных дает возможность снизить используемые для хранения ресурсы памяти ЭВМ, время, необходимое для передачи информации, а также, сокращение времени решения дальнейших задач.

Рассмотрим алгоритмы компрессии массивов точек, полученных при решении задачи Коши численным методом. При решении жестких систем обыкновенных дифференциальных уравнений (ОДУ) классическими методами возникают сложности, связанные с мелким шагом. Увеличение шага приводит к росту погрешности. Обработка всех вычисленных данных оказывается обременительной по необходимому объему памяти и времени. Задача уменьшения объема вычисленных данных при сохранении особенностей решения является актуальной.

Существуют различные подходы к прореживанию данных. Наиболее часто используется алгоритм прореживания данных за счет выборки расчетных точек, отстоящих друг от друга на заданное число шагов интегрирования, определяемым пользователем. При использовании такого подхода может быть потерян характер поведения решения (всплески), поэтому он может использоваться только для гладких решений.

Далее рассмотрим методы, основанные на различных критериях отсева точек, и сравним их между собой и с алгоритмом, основанным на выборке расчетных точек, отстоящих друг от друга на заданное число шагов интегрирования.

Алгоритм Компрессии 1

Пусть мы имеем дискретную функцию $y[x_i]$, заданную на сетке $x_0, x_1, ..., x_n$, с равномерным шагом h, полученную в результате численного решения задачи Коши методом Рунге-Кутты.

Вычисление сжатого множества точек будем проводить по следующему алгоритму:

В начале положим, что сжатое множество равно исходному и зададим числовое значение критерия ε.

1) Для точки 0 и 2 вычисляем

$$H_0 = \frac{\left|(y_2 - y_0)x_1 - hay_1 - x_0 y_2 + y_0 x_2\right|}{\sqrt{(y_2 - y_0)^2 + h^2 a^2}}. \quad (1)$$

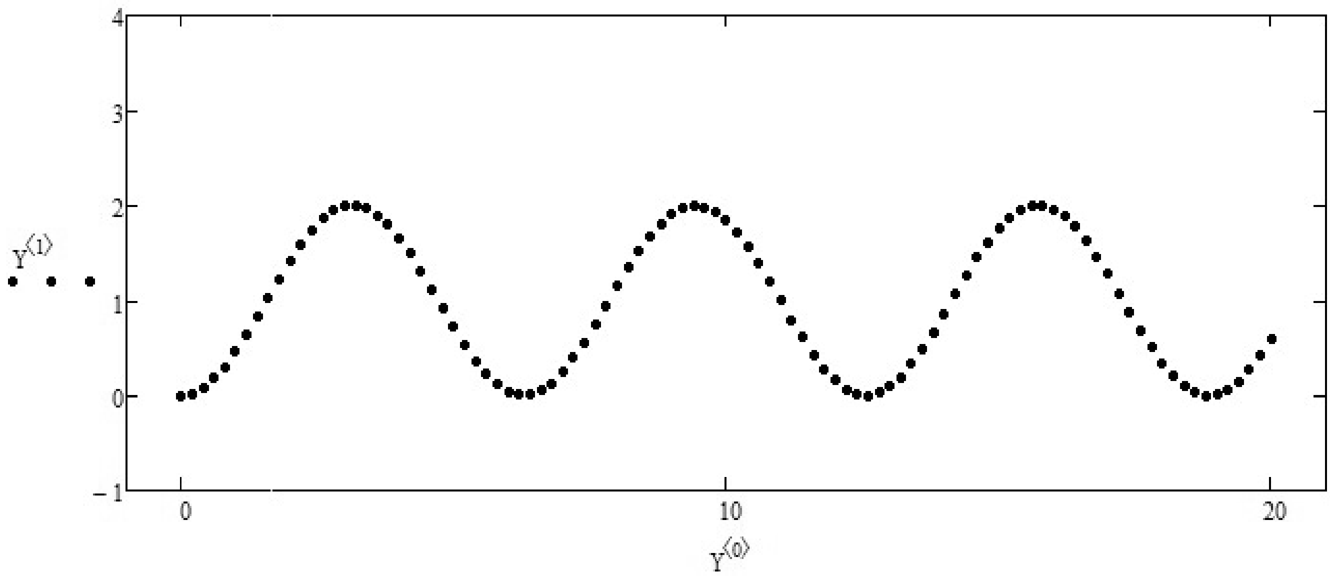

Рис. 1. Графическое изображение исходного сигнала (100 точек) .

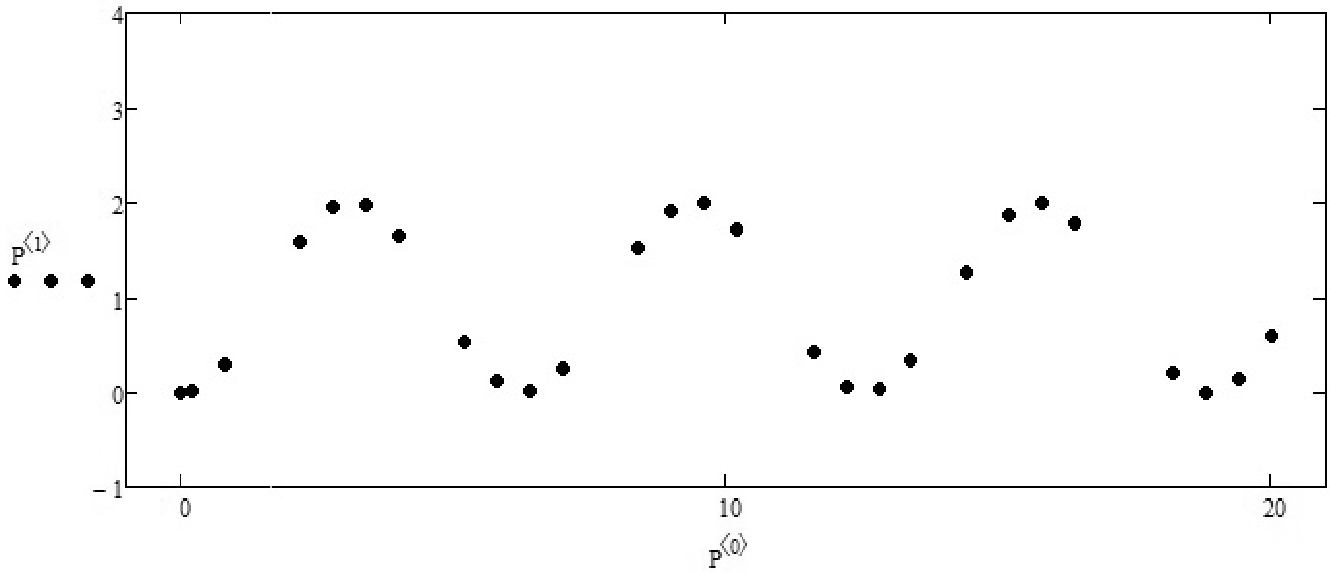

Рис. 2. Графический результат применения Алгоритма 1 (27 точек, $\varepsilon = 0.04$).

Если $\varepsilon > H_0$, то значение в точке 1 можно исключить, в противном случае, элемент оставляем.

2) Для точки 0 и 3 вычисляем

$$H_0 = \frac{\left|(y_3 - y_0)x_1 - hay_1 - x_0 y_3 + y_0 x_3\right|}{\sqrt{(y_3 - y_0)^2 + h^2 a^2}} \qquad (2)$$

и

$$H_1 = \frac{\left|(y_3 - y_0)x_2 - hay_2 - x_0 y_3 + y_0 x_3\right|}{\sqrt{(y_3 - y_0)^2 + h^2 a^2}}. \qquad (3)$$

Если $\varepsilon > H_0$ и $\varepsilon > H_1$, то значения в точках 1 и 2 можно исключить, в противном случае, элементы остается.

...

n) Для точки 0 и n вычисляем

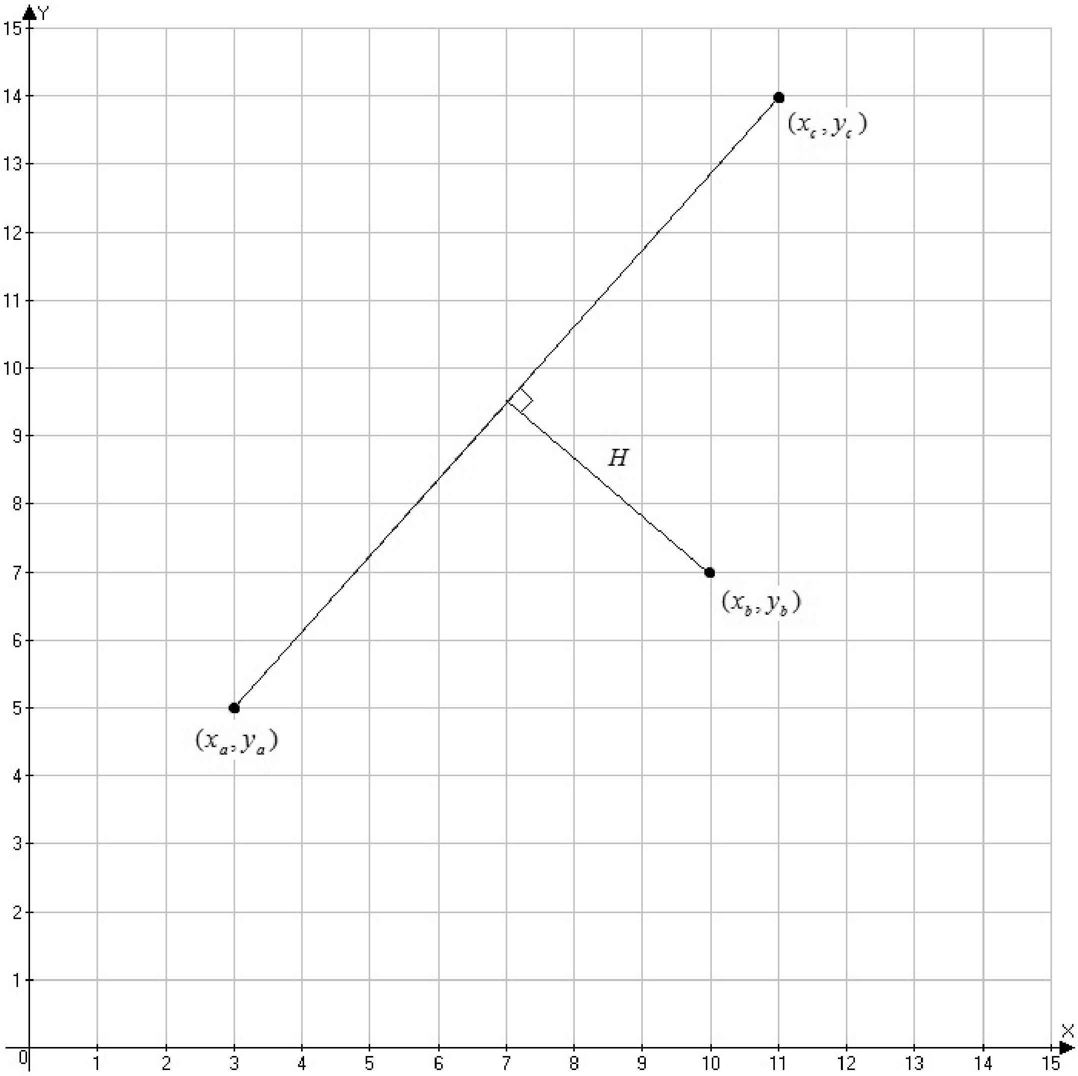

Рис. 3. Перпендикуляр от точки (x_b, y_b) к прямой заданной точками (x_a, y_a) и (x_c, y_c).

$$H_i = \frac{|(y_n - y_0)x_i - hay_i - x_0 y_n + y_0 x_n|}{\sqrt{(y_n - y_0)^2 + h^2 a^2}}. \quad (4)$$

Если $\varepsilon > H_i$ для всех i, то значения $1, \ldots, n-1$ можно исключить, в противном случае, элементы остаются.

n+1) Находим из n вариантов вариант в котором можно исключить наибольшее количество точек $1, \ldots, k-1$. Затем, повторяем алгоритм, в качестве начальной точки приняв k.

Алгоритм повторяется, пока k не будет равно n.

Данный алгоритм повторяется для каждой из n точек приняв ее за нулевую, и проверка осуществляется в обе стороны. После чего мы имеем n вариантов сокращенных множеств из которых выберем множество минимальной длины.

В результате выполнения указанной последовательности действий, мы получим сжатое множество. Изменяя значение величины ε-критерия, можно управлять числом элементов вошедших в сжатое множество, от этого будет зависеть количество исключаемой информации и степень сжатия.

Основная формула:

$$H_i = \frac{|(y_{k+a} - y_k)x_i - hay_i - x_k y_{k+a} + y_k x_{k+a}|}{\sqrt{(y_{k+a} - y_k)^2 + h^2 a^2}} \text{ —}$$

для обработки вправо. (5)

$$\widetilde{H}_i = \frac{|(y_{k-a} - y_k)x_i - hay_i - x_k y_{k-a} + y_k x_{k-a}|}{\sqrt{(y_{k-a} - y_k)^2 + h^2 a^2}} \text{ —}$$

для обработки влево. (6)

a — отрезок, на котором осуществляется вычисление высот для его внутренних точек, на данном этапе,

k — точка, являющаяся начальной в данный текущий момент,

i — номер точки, для которой осуществляется вычисление высоты.

На рисунке 1 и 2 показан пример исходного сигнала и сжатого.

Иллюстрация рассматриваемого расстояния H_i

Найдем расстояние от точки (x_b, y_b) до прямой заданной точками (x_a, y_a) и (x_c, y_c).

Составим уравнение прямой:

$$\frac{x - x_a}{x_c - x_a} = \frac{y - y_a}{y_c - y_a}. \tag{7}$$

Сначала найдем расстояние от начала координат до прямой:

$$(y_c - y_a)x + (x_a - x_c)y - x_a y_c + y_a x_c = 0. \tag{8}$$

Приведем данное уравнение к виду уравнения прямой с угловым коэффициентом

$$y = -\frac{y_c - y_a}{x_a - x_c}x - \frac{-x_a y_c + y_a x_c}{x_a - x_c} \tag{9}$$

отсюда видно, что угловой коэффициент

$$k = -\frac{y_c - y_a}{x_a - x_c}. \tag{10}$$

Уравнение прямой, проходящей через начало координат перпендикулярно данной прямой имеет вид

$$y = \frac{x_a - x_c}{y_c - y_a}x. \tag{11}$$

Из системы уравнений

$$\begin{cases} y = \dfrac{x_a - x_c}{y_c - y_a}x \\ (y_c - y_a)x + (x_a - x_c)y - x_a y_c + y_a x_c = 0 \end{cases} \tag{12}$$

можно найти координаты точки $M(x,y)$, которая является пересечением данной прямой и перпендикуляра опущенного на неё.

$$x = \frac{-(y_c - y_a)(-x_a y_c + y_a x_c)}{(y_c - y_a)^2 + (x_a - x_c)^2} \tag{13}$$

$$y = \frac{-(x_a - x_c)(-x_a y_c + y_a x_c)}{(y_c - y_a)^2 + (x_a - x_c)^2} \tag{14}$$

Расстояние от начала координат (0,0) до $M(x,y)$ и есть искомое расстояние от данной прямой до начала координат:

$$H = \sqrt{x^2 + y^2} =$$

$$= \sqrt{\frac{(y_c - y_a)^2(-x_a y_c + y_a x_c)^2}{((y_c - y_a)^2 + (x_a - x_c)^2)^2} + \frac{(x_a - x_c)^2(-x_a y_c + y_a x_c)^2}{((y_c - y_a)^2 + (x_a - x_c)^2)^2}} =$$

$$= \frac{|-x_a y_c + y_a x_c|}{\sqrt{(y_c - y_a)^2 + (x_a - x_c)^2}} \tag{16}$$

Расстояние от произвольной заданной точки $A(x,y)$ до данной прямой осуществляется при помощи параллельного переноса осей координат, приняв за начало координат точку $A(x,y)$. То есть

$$\begin{cases} x = x + x_b \\ y = y + y_b \end{cases} \tag{17}$$

И, следовательно, уравнение данной прямой в новой системе координат будет иметь вид

$$(y_c - y_a)(x + x_b) + (x_a - x_c)(y + y_b) -$$

$$- x_a y_c + y_a x_c = 0 \tag{18}$$

или

$$(y_c - y_a)x + (x_a - x_c)y - x_a y_c + y_a x_c +$$

$$+ (y_c - y_a)x_b + (x_a - x_c)y_b = 0 \tag{19}$$

В новой системе координат точка $A(x,y)$ является началом координат. Значит, расстояние H можно найти по формуле:

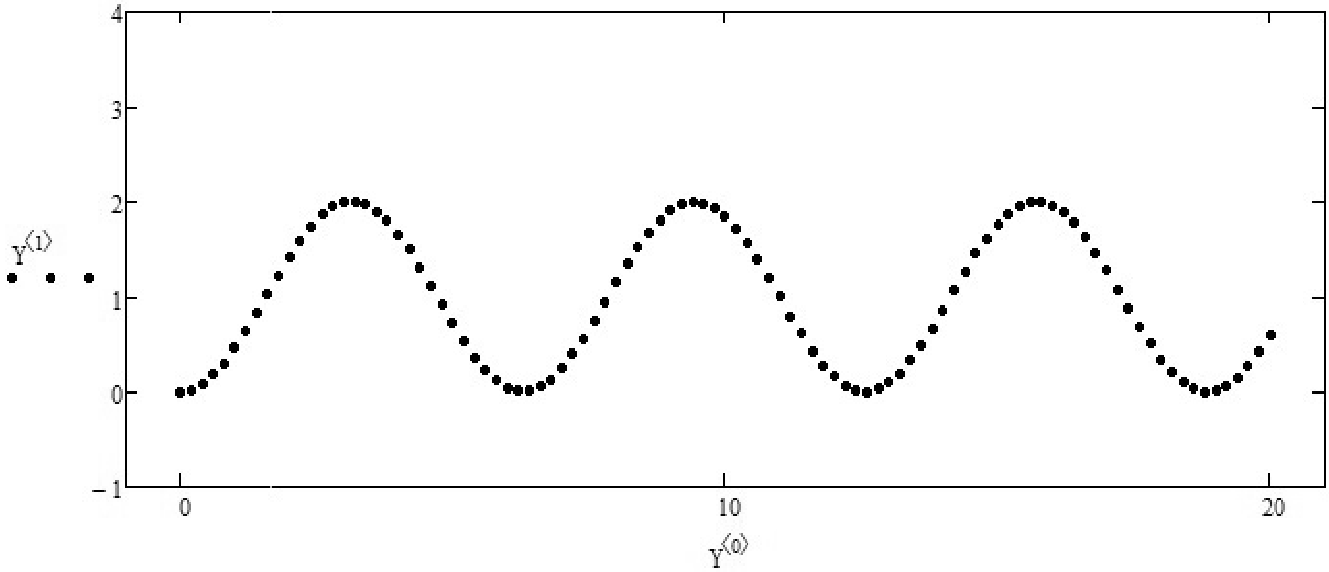

Рис. 4. Графическое изображение исходного сигнала

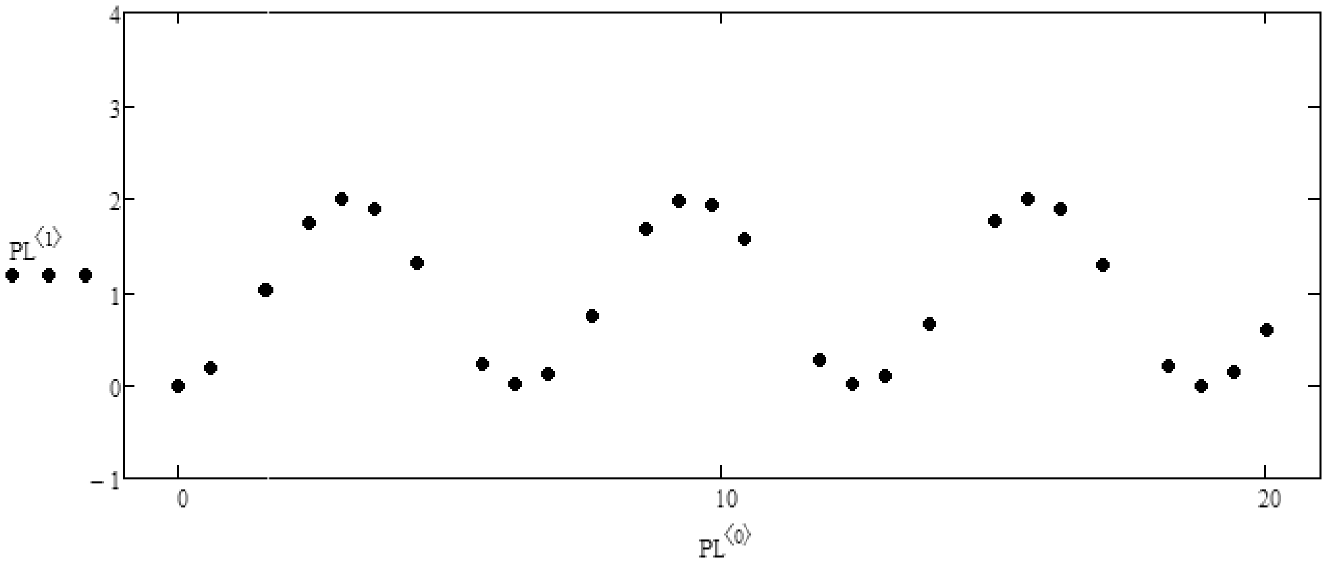

Рис. 5. Графический результат применения облегченного Алгоритма 1 (27 точек, $\varepsilon = 0.044$).

$$H = \frac{\left| -x_a y_c + y_a x_c + (y_c - y_a)x_b + (x_a - x_c)y_b \right|}{\sqrt{(y_c - y_a)^2 + (x_a - x_c)^2}} \qquad (20)$$

(Такая схема для удобства реализации программы).

Алгоритм Компрессии 1 (упрощенный вариант)

Для большого количества точек алгоритм 1 является слишком сложным и следует использовать вариант с той же самой формулой расчета высоты H, но с облегченным поиском точек, которые следует исключить.

1) Для точки 0 и 2 вычисляем

$$H_1 = \frac{\left|(y_2 - y_0)x_1 - hay_1 - x_0 y_2 + y_0 x_2\right|}{\sqrt{(y_2 - y_0)^2 + h^2 a^2}}. \qquad (21)$$

Если $\varepsilon > H_1$, то значение 1 можно исключить, и перейти к пункту 2. В противном случае, за начальный элемент принимается точка 1.

2) Для точки 0 и 3 вычисляем

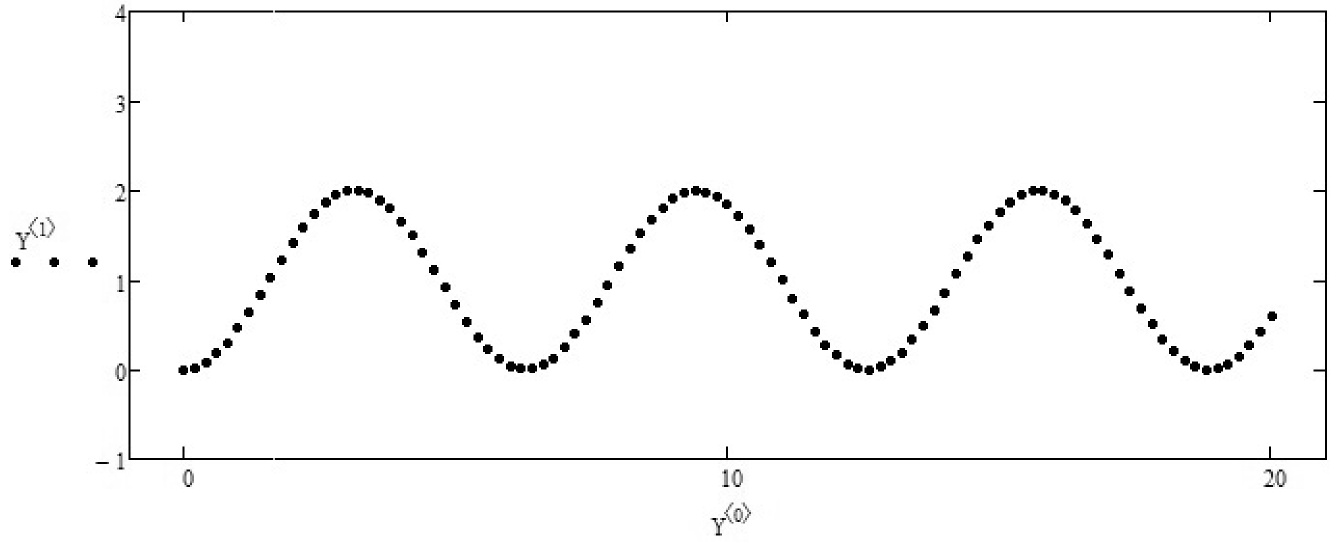

Рис. 6. Графическое изображение исходного сигнала

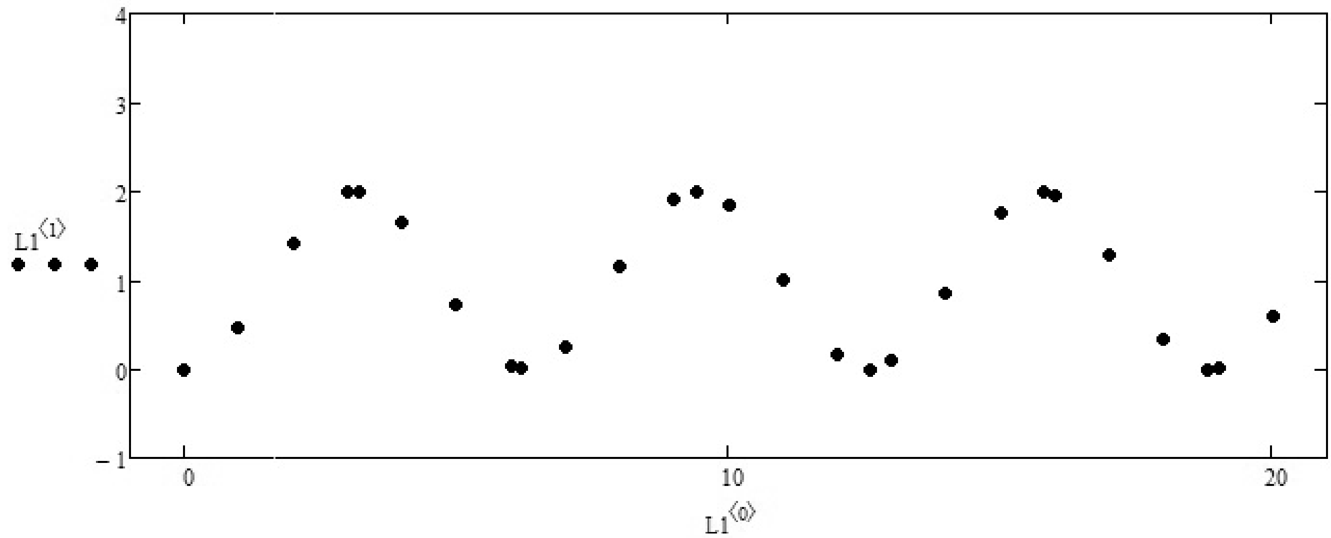

Рис. 7. Графический результат применения облегченного Алгоритма 3 (27 точек, 20 интервалов)

$$H_1 = \frac{|(y_3 - y_0)x_1 - hay_1 - x_0 y_3 + y_0 x_3|}{\sqrt{(y_3 - y_0)^2 + h^2 a^2}} \tag{22}$$

и

$$H_2 = \frac{|(y_3 - y_0)x_2 - hay_2 - x_0 y_3 + y_0 x_3|}{\sqrt{(y_3 - y_0)^2 + h^2 a^2}}. \tag{23}$$

Если $\varepsilon > H_1$ и $\varepsilon > H_2$, то значения 1 и 2 можно исключить, и перейти к пункту 3.

в противном случае, элементы остаются. Далее переходим к пункту k.

...

s) Для точки 0 и S вычисляем

$$H_i = \frac{|(y_s - y_0)x_i - hay_i - x_0 y_s + y_0 x_s|}{\sqrt{(y_s - y_0)^2 + h^2 a^2}}. \tag{24}$$

Если условие $\varepsilon > H_i$ не является истинным для всех i, то значения $1,\ldots,s-1$ можно исключить, в противном случае, элементы остаются.

k) Затем, повторяем алгоритм, в качестве начальной точки приняв последнюю точку из пункта $k-2$.

Алгоритм повторяется, пока k не будет равно n.

Таким образом, мы находим сокращенное множество.

Основная формула:

$$H_i = \frac{|(y_{k+a} - y_k)x_i - hay_i - x_k y_{k+a} + y_k x_{k+a}|}{\sqrt{(y_{k+a} - y_k)^2 + h^2 a^2}} \quad (25)$$

a — отрезок, на котором осуществляется вычисление высот для его внутренних точек, на данном этапе.

k — точка, являющаяся начальной в данный текущий момент.

i — номер точки, для которой осуществляется вычисление высоты.

На рисунке 4 и 5 показан пример исходного набора точек и сжатого набора для упрощенного алгоритма компрессии 1.

Алгоритм Компрессия 3 (min и max на отрезке)

Как и в двух предыдущих алгоритмах, имеем дискретную функцию $y[x_i]$, заданную на сетке $x_0, x_1, \ldots, x_n$, с равномерным шагом h. Разобьем весь интервал интегрирования на подинтервалы одинаковой длины. Можно задать количество подинтервалов интегрирования или длину подинтервала интегрирования.

Рассмотрим три алгоритма прореживания данных:

1. Из каждого подинтервала брать только его крайние точки.
2. Точки, имеющие максимальное и минимальное значение на данном подинтервале.
3. Объединим 1 и 2: крайние точки и точки, имеющие максимальное и минимальное значение.

Третий комбинированный метод является оптимальным и уменьшает вероятность потери данных.

Изменяя количество подинтервалов, можно управлять числом элементов, вошедших в сжатое множество.

Заключение

В результате работы, можно провести сравнение действия алгоритмов. После выполнения 2 алгоритма компрессии точки распределены по кривой относительно равномерно, а в алгоритме 1 в основном сосредоточены в местах максимальной кривизны кривой, следовательно, на графике мы получаем более плавные переходы, что дает наиболее полное представление о функции. Если рассматривать алгоритмы с точки зрения реализации и вычислительной сложности, то оптимальным является второй алгоритм. Время работы программы самое короткое. К тому же он достаточно хорошо передает «всплески» решения.

Упрощенная версия алгоритма 1 в данном случае, дают почти такие же результаты, в отдельных случаях, абсолютно точно такие же, хотя время работы программы значительно уменьшается. То есть, упрощенная версия 1 алгоритма является наиболее удачной.

Все данные алгоритмы не эффективны для шумоподобных сигналов и отдельных частных случаях.

ЛИТЕРАТУРА

1. Шнейдер В.Е., Слуцкий А.И., Шумов А.С. Краткий курс высшей математики. М.: Высшая школа, 1972. — 640 с
2. Пискунов Н.С. Дифференциальное и интегральное исчисления. В 2-х т. Т.II: М.: Интеграл-Пресс, 2005. — 544 с.
3. Пирумов У.Г. Численные методы: Учеб. пособие для вузов /. — 2-е изд., испр. и доп. М.: Дрофа, 2003. — 221 с.
4. Керниган Б., Ритчи Д. Язык программирования С М.: Вильямс, 2013. — 304 с.

© Брусова Анна Александровна (SupernaturalAnn@yandex.ru).
Журнал «Современная наука: актуальные проблемы теории и практики»