ISSN 2074-7888, Наукові праці Донецького національного технічного університету, серія Проблеми моделювання та автоматизації проектування, 2011, Випуск 10 (197)

УДК 519.65

А.Б. Иващенко (ассистент),
В.Н. Беловодский (канд. техн. наук, доц.)
Донецкий национальный технический университет
alesya_iva@list.ru, belovodskiy@cs.dgtu.donetsk.ua

НЕКОТОРЫЕ ВАРИАЦИИ МЕТОДА ЭГЛАЙСА СИНТЕЗА АППРОКСИМИРУЮЩИХ ФУНКЦИЙ

Работа посвящена обсуждению ряда вариантов изменения базового алгоритма метода синтеза аппроксимирующих функций, предложенного В. Эглайсом в конце 70-х годов прошлого века. С помощью программ, реализующих предложенные модификации, и серии вычислительных экспериментов проведен сравнительный анализ результатов использования модифицированных алгоритмов с исходным.

Ключевые слова: аппроксимация, модификация, базовый алгоритм, регрессионная модель, зависимость, банк функций, элиминация, корреляция остатков с откликом.

Введение

В работе [1] изложены результаты разработки и отладки программы, реализующей метод синтеза аппроксимирующих функций, предложенный Эглайсом в 70-х годах прошлого века [2].

Этот метод представляет собой комплексный подход к исследованию сложных систем на ЭВМ и помогает формировать регрессионные модели для их анализа. Он позволяет, на основе экспериментальных данных, проводить восстановление, как структуры модели, так и ее параметров.

При восстановлении модели данный алгоритм позволяет определять ее состав и структуру на основании конкретных внутренних критериев. Определение структуры и значений неизвестных параметров (коэффициентов) дает возможность судить о характере причинно-следственных связей, присущих анализируемой системе. Таким образом, восстановленная модель позволит также судить о виде и степени взаимодействия между независимыми и зависимыми факторами, о месте и значении каждого элемента в общем процессе функционирования системы.

Данный алгоритм универсален тем, что позволяет строить модели для анализа большого числа разнообразных задач. Причем для каждой конкретной задачи структура модели генерируется таким образом, чтобы модель оптимально соответствовала исследуемой системе и отражала те аспекты и связи в системе, которые являются существенными в данной зада

че. Поэтому данный алгоритм характеризуется принципами самоорганизующихся и самонастраивающихся систем.

Этап элиминации, в котором происходит удаление несущественных функции и окончательный подбор уравнения регрессии, позволяет реализовать принципы непротиворечивости элементов и абстрагирования от второстепенных деталей, то есть первоначально, сформировав достаточно точную модель, а значит, чаще всего, и сложную, проследить за изменениями точности модели, которые происходят в результате удаления несущественных функций и, в конечном счете, избавиться лишь от тех из них, присутствие которых незначимо для точности модели, тем самым упростив модель.

Разработчики моделей находятся под действием двух взаимно противоречивых тенденций: стремления к полноте описания исследуемой системы и стремления к получению требуемых результатов возможно более простыми средствами. Модели по своей природе всегда носят приближенный характер. Возникает вопрос, каким должно быть это приближение. Чтобы отразить все сколько-нибудь существенные свойства, модель необходимо детализировать. С другой стороны, строить модель, приближающуюся по сложности к реальной системе, очевидно, не имеет смысла. Она не должна быть настолько сложной, чтобы нахождение решения оказалось слишком затруднительным. Компромисс между этими двумя требованиями достигается нередко путем проб и ошибок.

В данном методе реализация компромисса, фактически, осуществляется двумя путями:

– во-первых, программная реализация данного метода позволяет пользователю осуществлять поиск компромиссной «золотой середины» самостоятельно, путем манипулирования параметрами модели. Ручное задание параметров (число перспективных функций, максимальная степень одночлена, точка элиминации) позволяет пользователю опытным путем определять оптимальную модель и производить корректировку модели, подбирая наиболее соответствующую требуемой точности;

– во-вторых, сама методика формирования модели построена на идее (принципе) компромисса между требуемой точностью результатов моделирования (или адекватностью, сложностью модели) и эффективностью решений (простотой модели), т.е. модели, восстанавливаемые в результате применения метода, носят компромиссный характер.

В ходе выполнения вычислительных экспериментов и анализа полученных результатов был сформулирован ряд предложений, касающихся изменения порядка выполнения отдельных этапов и направленных на улучшение этого метода. Некоторые из этих предложений и обсуждаются ниже.

Базовый алгоритм

Концептуально идея базового алгоритма заключается в следующем. Аппроксимирующая функция строится в классе степенных разложений заданного порядка и процесс ее нахождения предполагает выполнение следующих этапов. На первом из них, по заданной степени одночлена, формируется множество базисных функций, из которого, с использованием двучленных регрессионных уравнений, выбирается заданное количество перспективных. После этого, направленным перебором производится элиминация – окончательное формирование функции, адекватной исходной информации. На рисунке 1 в общем виде представлена схема работы программы, реализующей этот алгоритм.

Перейдем к изложению предложенных изменений.

Рисунок 1 – Принцип работы алгоритма базовой версии программы

Предложение 1. Формирование набора перспективных функций проводить по корреляционному признаку

Данное предложение касается корректировки принципа отбора базисных функций в число перспективных и обусловлено следующими обстоятельствами. Дело в том, что в ряде случаев, и это было замечено в ходе вычислительных экспериментов, отдельные функции, заведомо присутствующие в структуре модели, не попадают в число перспективных, оказываясь буквально в конце списка функций-претендентов, упорядоченного по убыванию «критерия перспективности», то есть суммы квадратов отклонений. Представляется вполне правдоподобным, что такое положение дел можно исправить или, хотя бы улучшить, если в список перспективных отбирать функции с максимальными коэффициентами корреляции.

Напомним [1], что в базовом алгоритме для каждой i-ой функции-предиктора определяются коэффициенты линейной регрессии, затем вычисляется сумма квадратов отклонений s_i. В программе, его реализующей, после формирования банка функций предусмотрено также формирование таблицы, хранящей в себе информацию о порядковом номере функции и другие дополнительные характеристики, а именно, соответствующие ей коэффициенты линейной регрессии уравнения и суммарные квадратичные ошибки s_i. Это удобно как для восстановления связи с набором степеней, с помощью которого была сформирована функция, так и для сортировки и упорядочения функций без потери информации о них. Далее, согласно алгоритму, осуществляется сравнение всех s_i между собой и упорядочение функций по возрастанию значения s_i. Отметим, что в данном случае, обработка происходит не в самом банке функций, а внутри информационной таблицы. Использование не самих функций, а как бы указателей на них, оправдано, в том числе, и тем, что минимизирует риск «обратиться не по адресу», т.к. в противном случае, после сортировок массива, удаления ненужных функций и, в результате этого, смещения всего массива, невозможно определить, где находится требуемая функция.

Таким образом, при описанной организации алгоритма, в начале таблицы оказываются указатели на функции с наименьшими s_i. Первые p из этих функций и отбираются в число перспективных.

Для отбора же функций по корреляционному признаку, предлагается для каждой из функций банка вместо суммы квадратов отклонений вычислять коэффициент корреляции самой функции и исходного отклика. Далее, – упорядочить список функций, но уже по убыванию абсолютного значения коэффициента корреляции. И, наконец, как и в базовой версии, верхние p из них отобрать в число перспективных.

Отметим, что коэффициент корреляции (парный коэффициент корреляции, коэффициент корреляции Пирсона) – характеризует степень тесноты линейной связи между случайными величинами X и Y. Он применяется тогда, когда данные наблюдений можно считать случайными и выбранными из генеральной совокупности, распределенной по многомерному нормальному закону. Выборочное значение *r* коэффициента корреляции рассчитывается по формуле:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \qquad (1)$$

Напомним, что значение *r* изменяется в пределах $-1 \le r \le 1$. При $|r| = 1$ этот коэффициент подтверждает функциональную линейную зависимость между величинами *X* и *Y,* если же переменные независимы, то *r*=0. Положительные значения коэффициента корреляции указывают на одинаковый характер тенденции взаимосвязанного изменения величин *X* и *Y* (например, увеличение *X* влечет и увеличение *Y),* отрицательные значения указывают на противоположную тенденцию. В случаях, если распределения величин *X* и *Y* отличаются от нормального или одна из величин не является случайной, коэффициент корреляции можно использовать лишь в качестве одной из возможных характеристик степени тесноты связи.

В пакете MatLab имеется встроенная функция вычисления коэффициента корреляции corr(x,y).

В ходе вычислительного эксперимента оказалось, что результаты отбора перспективных функций по коэффициенту корреляции оказались практически идентичны с результатами отбора функций по суммарным квадратам остатков. Для демонстрации этого факта на рисунке 2 показан список функций с соответствующими им суммами квадратов отклонений и коэффициентами корреляции. Здесь и далее, для иллюстрации, рассматривается пример восстановления функции

$$y = 3x_1^3 + 2x_2^2 + x_1^{-1}x_2^2 + 6x_1x_2 + 7x_2 + 5. \qquad (2)$$

Отметим, что в случае, когда данные предварительно масштабируются, коэффициенты корреляции изменяются, как, впрочем, и суммы квадратов отклонений, но после упорядочивания последовательность расположения функций в таблице по корреляциям и по суммам квадратов сохраняется (рисунок 3).

Объяснить данное явление, по-видимому, можно так (хотя для посвященных в математическую статистику это, наверное, очевидно). Значение *r* является измерителем степени тесноты именно линейной

статистической связи между переменными. В то же время, сумма квадратов отклонений как раз и представляет собой ошибку в предсказании откликов с помощью все той же линейной регрессии, и, очевидно, чем выше коэффициент корреляции, тем более правдоподобна линейная зависимость и, следовательно, меньше общая сумма квадратов отклонений.

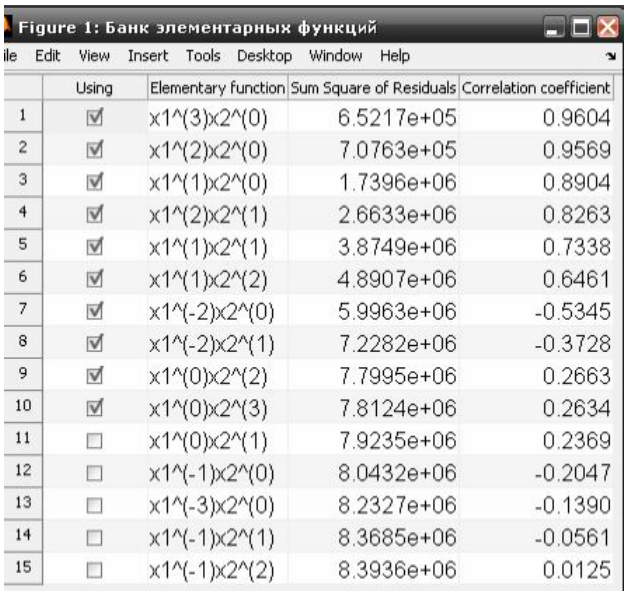
Figure 1: Банк элементарных функций

ile Edit View Insert Tools Desktop Window Help

	Using	Elementary function	Sum Square of Residuals	Correlation coefficient
1	☑	x1^(3)x2^(0)	6.5217e+05	0.9604
2	☑	x1^(2)x2^(0)	7.0763e+05	0.9569
3	☑	x1^(1)x2^(0)	1.7396e+06	0.8904
4	☑	x1^(2)x2^(1)	2.6633e+06	0.8263
5	☑	x1^(1)x2^(1)	3.8749e+06	0.7338
6	☑	x1^(1)x2^(2)	4.8907e+06	0.6461
7	☑	x1^(-2)x2^(0)	5.9963e+06	-0.5345
8	☑	x1^(-2)x2^(1)	7.2282e+06	-0.3728
9	☑	x1^(0)x2^(2)	7.7995e+06	0.2663
10	☑	x1^(0)x2^(3)	7.8124e+06	0.2634
11	☐	x1^(0)x2^(1)	7.9235e+06	0.2369
12	☐	x1^(-1)x2^(0)	8.0432e+06	-0.2047
13	☐	x1^(-3)x2^(0)	8.2327e+06	-0.1390
14	☐	x1^(-1)x2^(1)	8.3685e+06	-0.0561
15	☐	x1^(-1)x2^(2)	8.3936e+06	0.0125

Рисунок 2 – Ранжирование функций по различным критериям

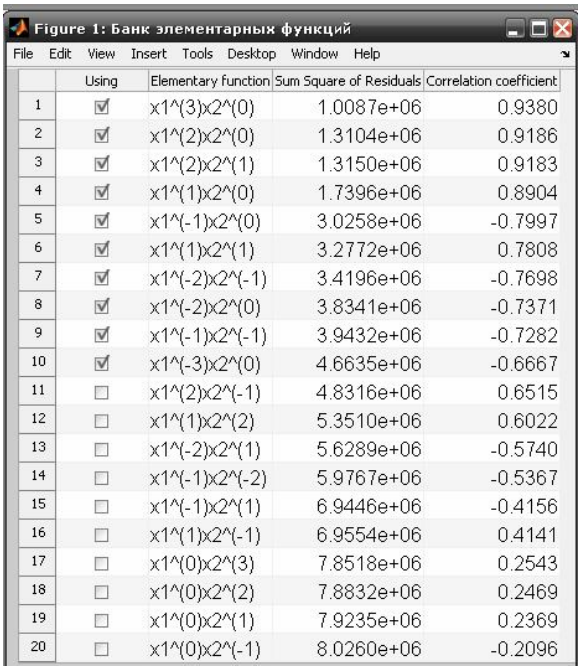
Figure 1: Банк элементарных функций

File Edit View Insert Tools Desktop Window Help

	Using	Elementary function	Sum Square of Residuals	Correlation coefficient
1	☑	x1^(3)x2^(0)	1.0087e+06	0.9380
2	☑	x1^(2)x2^(0)	1.3104e+06	0.9186
3	☑	x1^(2)x2^(1)	1.3150e+06	0.9183
4	☑	x1^(1)x2^(0)	1.7396e+06	0.8904
5	☑	x1^(-1)x2^(0)	3.0258e+06	-0.7997
6	☑	x1^(1)x2^(1)	3.2772e+06	0.7808
7	☑	x1^(-2)x2^(-1)	3.4196e+06	-0.7698
8	☑	x1^(-2)x2^(0)	3.8341e+06	-0.7371
9	☑	x1^(-1)x2^(-1)	3.9432e+06	-0.7282
10	☑	x1^(-3)x2^(0)	4.6635e+06	-0.6667
11	☐	x1^(2)x2^(-1)	4.8316e+06	0.6515
12	☐	x1^(1)x2^(2)	5.3510e+06	0.6022
13	☐	x1^(-2)x2^(1)	5.6289e+06	-0.5740
14	☐	x1^(-1)x2^(-2)	5.9767e+06	-0.5367
15	☐	x1^(-1)x2^(1)	6.9446e+06	-0.4156
16	☐	x1^(1)x2^(-1)	6.9554e+06	0.4141
17	☐	x1^(0)x2^(3)	7.8518e+06	0.2543
18	☐	x1^(0)x2^(2)	7.8832e+06	0.2469
19	☐	x1^(0)x2^(1)	7.9235e+06	0.2369
20	☐	x1^(0)x2^(-1)	8.0260e+06	-0.2096

Рисунок 3 – Ранжирование функций по различным критериям (после предварительного масштабирования данных)

Отметим, что до конца, пока, осталось непонятным, почему в обоих случаях функция $x_1^{-1}x_2^2$ каждый раз оказывается в конце списка, будучи, на самом деле, существенной для построения точной модели. Впрочем,

парный коэффициент корреляции оценивает связь только двух переменных, без учета влияния или наложения других факторов, и по этой причине опрометчиво было ожидать, что он должен быть близок к единице. Вместе с тем, обращает на себя внимание весьма низкий ее коэффициент корреляции (0.0125), в результате чего она оказывается даже менее «привлекательной», чем любая другая из неперспективных функций.

В заключение отметим, что хотя рассмотренная здесь модификация алгоритма и не привела к улучшению рассматриваемого метода, тем не менее, становится понятным, что использование коэффициента корреляции при отборе перспективных функций также вполне приемлемо.

Предложение 2. Формирование набора перспективных функций проводить по «остаточному» признаку.

Вариант 1. Заметим, что в базовой версии алгоритма значение параметра p (количество функций-кандидатов, отбираемое программой в число перспективных) задает пользователь. При этом он обычно руководствуется своими субъективными предположениями о том, какое значение параметра p окажется достаточным, либо проводит серию экспериментов, перебирая различные варианты, из осторожности останавливаясь на излишне большом значении p. Следует отметить, что реализация такого перебора, т.е. экспериментальный подбор параметра p, увеличивает общее время обработки исходных данных и выполнение процесса элиминации, в частности, а, если же от него отказаться, то сохраняется элемент субъективности. Чтобы исключить указанные факторы из процесса аппроксимации, можно попробовать вообще обойтись без этого параметра, автоматически включая все функции банка в число перспективных, однако, безусловно, существенное увеличение затрат машинного времени удерживает от этого шага. В связи с этим возникает вопрос: существует ли формализованный способ определения достаточного числа перспективных функций?

Один из вариантов формирования перспективного множества представляется в виде учета корреляционной связи между остатками (то есть той частью отклика, которую не удалось описать с помощью отобранных на данный момент функций) и функциями банка, которые еще не внесены в список перспективных. Действительно, логично допустить, что если существует высокая связь между остатками и некоторой функцией, то ее добавление в регрессионную модель вполне разумно, что позволит уменьшить ошибки и точнее описать экспериментальные данные.

Еще раз подчеркнем, что в базовом алгоритме число p задается пользователем и после ранжирования функций по возрастанию СКО (суммы квадратов отклонений) или же по убыванию коэффициента

корреляции, согласно рассмотренной выше модификации базового алгоритма, первые p функций автоматически включаются в число перспективных.

В данном же случае, предлагается первоначально ранжировать все функции банка в порядке возрастания СКО и полученный порядок их расположения сохранять до полного завершения работы алгоритма. Тем самым фиксируется порядок просмотра функций при отборе их в перспективные, а критерием включения очередной функции в число перспективных является величина корреляционной связи еще незадействованных функций с образующимися остатками. А именно, проводя пересчет коэффициентов регрессии после включения очередной функции, вычисляется разность значений (это и есть остатки или погрешности) между табличными и восстановленными откликами, и оценивается корреляционная связь этих остатков с функциями, которые еще не вошли в список перспективных. Если среди коэффициентов корреляции встречаются большие значения (сильная корреляция), то это свидетельствует о наличии перспективных функций среди неотобранных, т.е. о том, что отобранных функций пока недостаточно, поскольку, в противном случае, т.е. при достаточном объеме сформированного набора перспективных функций для полного описания исходных данных, следует ожидать слабой корреляции остальных элементов банка функций с образующимися к этому времени остатками. Когда же корреляция остатков с невключенными функциями уменьшается и становится незначительной, т.е. корреляция становится слабой, отбор перспективных функций прекращается и, таким образом, определяется достаточное число перспективных функций p. Отметим, что в таком случае, необходимо назначать новый управляющий параметр, определяющий контрольную или пороговую величину корреляции. Предполагается, что он будет задаваться экспериментатором и освободит его от задания параметра p. Будем называть его «предельно допустимой корреляцией остатков с функциями» и обозначать символом pdk. Не вдаваясь в детали отдельных этапов, покажем схематично принципиальную разницу базового и нового алгоритма. На рисунке 1 была представлена схема алгоритма базовой программы, а суть предлагаемых изменений продемонстрирована на рисунке 4. Обратим внимание, что порядок анализируемых функций, при включении их в число перспективных, в данном случае сохраняется таким же, как и в базовом алгоритме, а параметр pdk служит лишь критерием «останова» и определения достаточного значения для параметра p.

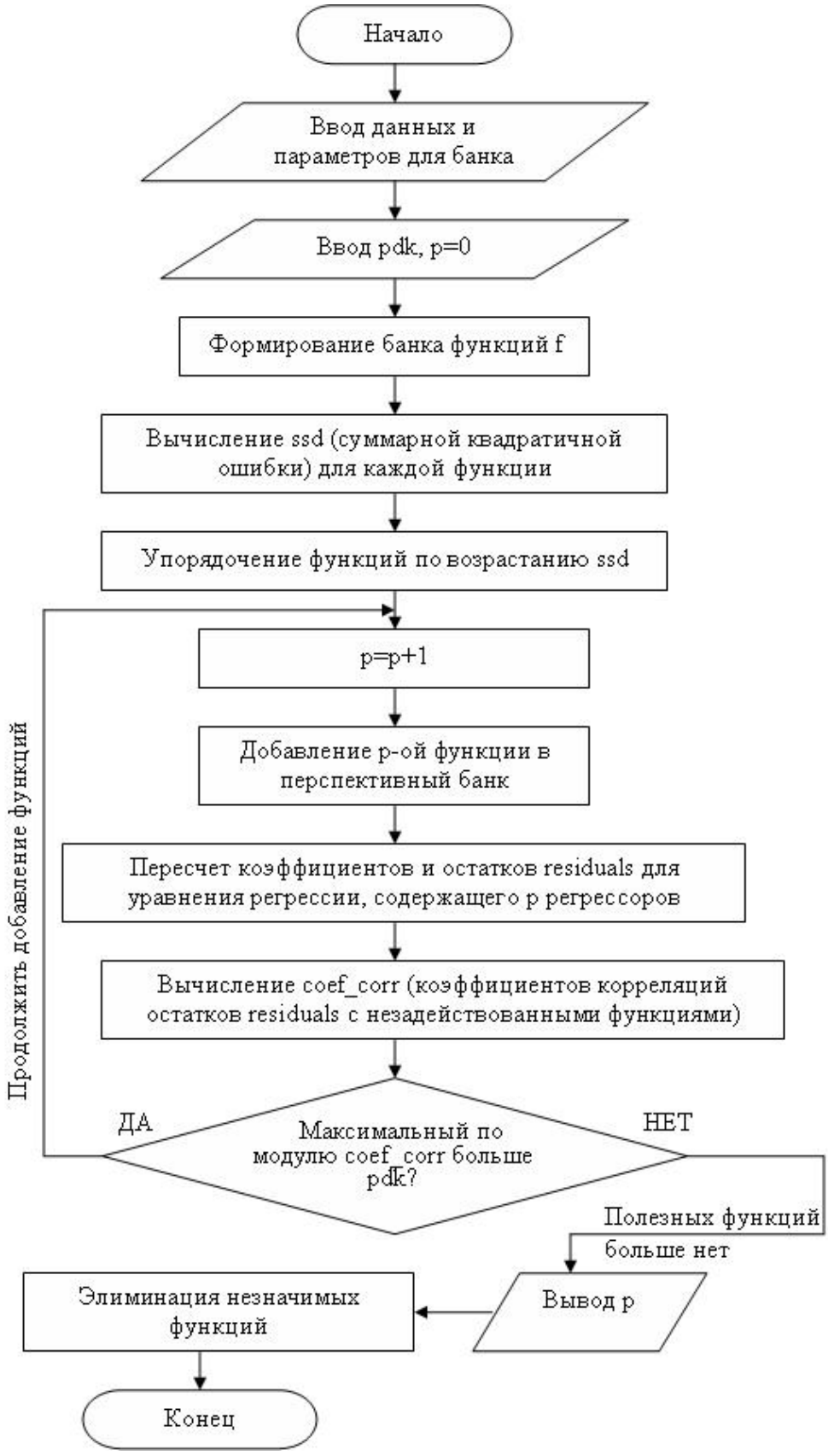

Рисунок 4 – Принцип работы алгоритма, реализующего Вариант 1

Укажем несколько замечаний в пользу предложенных изменений.

Известно, что коэффициент корреляции лежит в диапазоне [-1;1] и чем ближе его значение к нулю, тем больше оснований считать, что исследуемые величины независимы. С другой стороны, решение о том, каким должен быть порядок точности модели, принимает пользователь. Но, задавая конкретное значение параметра p, даже если оно вполне достаточное по его предположениям, он не может рассчитывать на включение всех «полезных» функций в число перспективных, поскольку сам по себе критерий перспективности необъективен.

В первой модификации показано, что суммарное квадратичное отклонение заменимо на коэффициент парной корреляции. На первый взгляд может показаться, что там же можно было бы отказаться и от параметра p, отбирая в перспективные лишь те функции, у которых коэффициент корреляции с откликом выше некоторого порогового значения, задаваемого пользователем. Но насколько объективным будет выбор такого порога пользователем? Ведь может случиться так, что коэффициент корреляции некоторой функции с откликом низкий лишь только потому, что его «вклад» в описание отклика невелик, но в «ансамбле» с другими функциями она могла бы дать подходящую регрессионную модель. Но, тогда, например, для восстановления функции (2), которая рассматривалась и ранее (рисунок 2), для нахождения требуемого решения этот порог должен был бы быть ниже 0,0125, что несколько абсурдно. Ведь задавая такой низкий корреляционный «проходной» порог, набор перспективных функций может получиться очень большим, что также является нежелательным. Фактически, это то же, что и задание пользователем заведомо большого значения параметра p. Поэтому такие соображения (задание пользователем порогового значения корреляции функций с откликом) должны быть отклонены.

Результаты тестирования рассматриваемого варианта алгоритма оказались несколько лучше предыдущего. Так, при восстановлении функции (2), для того, чтобы в перспективные попали все необходимые функции, достаточно положить $pdk = 0.02$ или меньше. В другом тестовом примере, а именно, при восстановлении зависимости $y = 6x_1x_2 + 7x_2 + 3x_1 + 7x_2^{-2} + x_1^{-1}x_2^{-1} + 5$, успешное включение всех функций достигалось уже при $pdk = 0.45$, хотя первоначальный коэффициент корреляции с откликом для «полезной» функции $x_1^{-1}x_2^{-1}$ был равен -0.188.

Таким образом, если пользователю вместо числа перспективных функций или пороговой корреляции функций с откликом предоставить задание значения *pdk* – предельно допустимого коэффициента корреляции их с остатками, то он может руководствоваться уже своими соображениями по поводу назначаемой степени точности. Так, например,

если пользователь укажет значение pdk =0.25, то пока есть хотя бы одна корреляционная связь остатков и функции, еще не попавшей в перспективные, превышающая 0.25 (то есть имеется возможность описать остатки с помощью оставшихся функций), в число перспективных на каждом шаге будет добавляться по одной функции (но в очередности убывания коэффициента их первоначальной корреляции с откликом). Когда корреляций незадействованных функций с остатками, превышающих заданное пользователем значение, не останется, то есть останутся лишь «незначимые» (по сравнению с установленным пользователем порогом) функции, т.е. с низкой описательной способностью по отношению к остаткам, тогда добавление функций прекращается. И число отобранных перспективных функций считается достаточным. Уменьшая значение pdk, пользователь получает возможность повышать точность аппроксимации за счет снижения допустимого порога связи конечных остатков с оставшимися функциями.

Безусловно, такой порядок формирования набора перспективных функций увеличивает трудоемкость алгоритма и общее операционное время, но, с другой стороны, избавляет пользователя от дополнительного экспериментирования при подборе значения параметра p и, на наш взгляд, повышает объективность его назначения.

Заметим, что предлагаемый алгоритм не меняет порядок включения функций, и последовательно включает функции в том порядке, который был определен первоначальным ранжированием по корреляционному признаку. Это означает, что в перспективные будут попадать также и те функции, у которых текущая корреляция с остатками уже ниже pdk, но по порядку вхождения в перспективные они находятся ближе, чем те, у которых она выше pdk и из-за которых, собственно говоря, и продолжается процесс добавления функций.

Вариант 2. Задача этой модификации – попытаться увеличить «скорость» отбора «полезных» функций в число перспективных.

Как уже отмечалось, в базовой версии программы и в рассмотренных выше модификациях порядок просмотра функций при включении их в число перспективных одинаков – он определяется предварительным ранжированием их по степени возрастания СКО. Поэтому сравнительный анализ предлагаемого варианта достаточно проводить с одной из них. С целью его сравнения с базовым алгоритмом, используем для определения объема перспективных функций параметр p (для сравнения с вариантом 1 необходимо было бы использовать pdk).

Еще раз подчеркнем, что во второй модификации величина коэффициентов корреляции остатков с незадействованными функциями не влияли на порядок, точнее, на последовательность включения функций в число перспективных, а лишь служила критерием их включения. Сами же

функции включались в том порядке, в котором они были упорядочены первоначально по критерию СКО (или, что, то же самое, по коэффициенту корреляции отклика с функциями). В блок-схеме рисунка 4 это этап «Упорядочение функций по возрастанию ssd».

В данной модификации предлагается изменить и установить следующий порядок включения базовых функций в число перспективных. А именно, после вычисления коэффициентов остаточной корреляции выделять незадействованную функцию, имеющую максимальную корреляционную связь с остатками. И затем, именно ее включать как перспективную. После этого заново пересчитывать коэффициенты регрессии, вычислять новые остатки, новые «остаточные» корреляции и определять следующую перспективную функцию. Таким способом определяется заданное число p перспективных функций. Не вдаваясь в детали, на рисунке 5 приведена принципиальная схема работы соответствующего алгоритма.

Покажем различие в результатах работы алгоритмов на примере нескольких тестов, указав перечень функций и проследив порядок их отбора с помощью базового и предлагаемого алгоритма.

Таблица 1 содержит результаты восстановления регрессионной зависимости $y = 3x_3^3 + 2x_2^2 + 1x_1 + 2x_1x_2x_3 + 5$ этими алгоритмами. В таблице приводятся функции, а точнее, соответствующие им наборы степеней, в порядке их включения в число перспективных, «полезные» наборы степеней выделены подчеркиванием. Можно заметить, что в базовом алгоритме последняя «полезная» функция будет включенной шестнадцатой по счету (алгоритм генерирует для этих данных всего 19 функций), а в предлагаемой модификации уже одиннадцатой.

Следует также отметить, что на порядок функций в том, и другом случае влияет качество исходных данных, например, число наблюдений. С ростом объема исходных данных «полезные» функции «быстрее» поднимаются к верхним строкам таблицы, а значит, и быстрее попадают в число перспективных. Это наглядно продемонстрировано в таблице 2 для той же функции $y = 3x_3^3 + 2x_2^2 + 1x_1 + 2x_1x_2x_3 + 5$, но с б**о**льшим числом наблюдений в исходных данных (91 вместо 20). Здесь для полного восстановления уравнения базовому алгоритму потребовалось бы задать параметр $p = 12$, а в Варианте 2 алгоритма,– всего лишь $p = 6$.

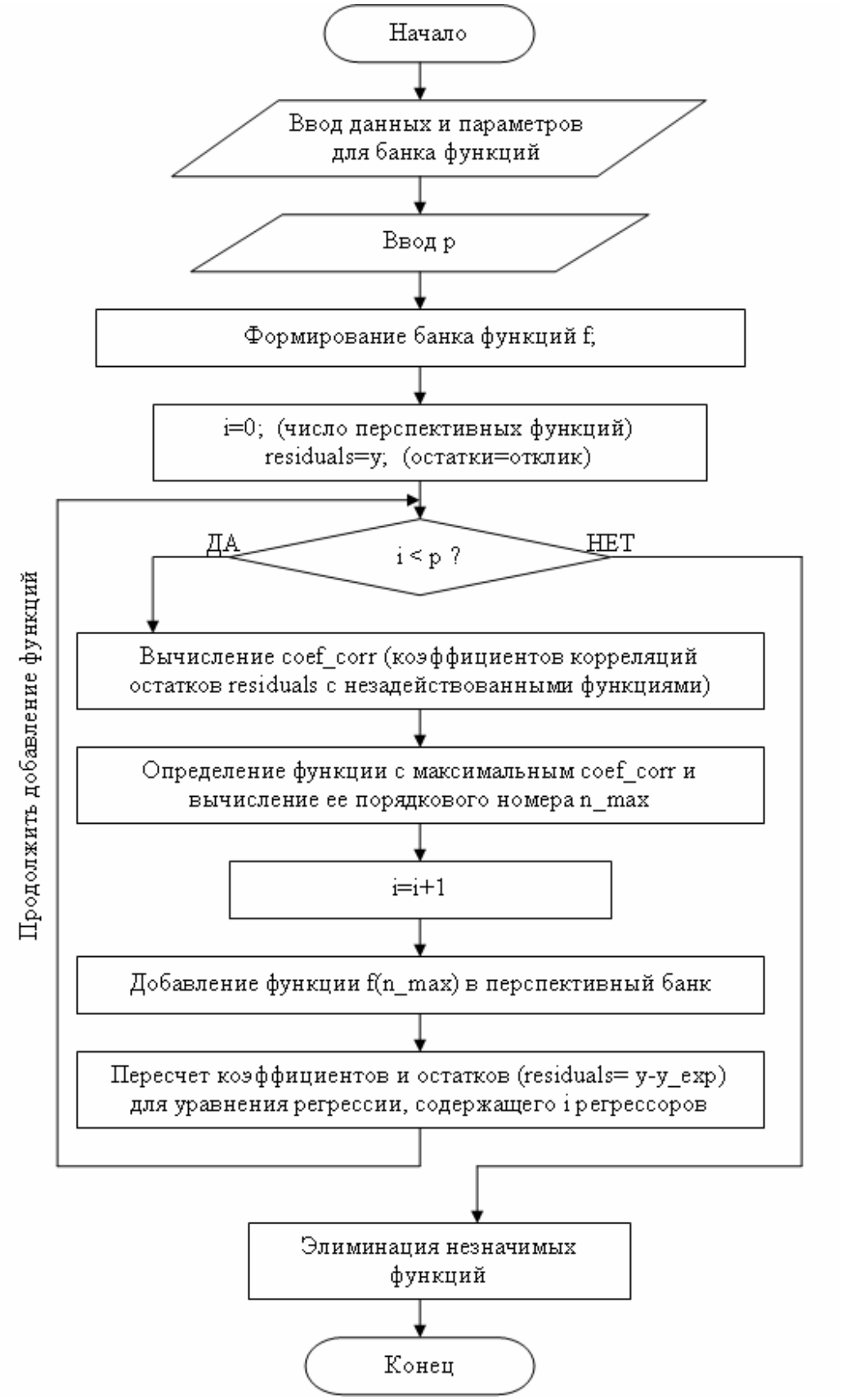

Рисунок 5 – Принцип работы алгоритма в Варианте 2

Таблица 1. Сравнение результатов алгоритмов на примере функции $y = 3x_3^3 + 2x_2^2 + 1x_1 + 2x_1x_2x_3 + 5$ (число наблюдений m=20)

№ п/п включения	Базовый алгоритм	Модификация 3
1	**0 0 3**	**0 0 3**
2	0 0 2	0 2 1
3	1 0 2	0 3 0
4	1 0 1	**1 1 1**
5	**0 0 1**	0 1 0
6	2 0 1	1 0 0
7	0 1 0	0 1 1
8	**0 2 0**	3 0 0
9	1 0 0	2 0 1
10	1 1 0	**0 0 1**
11	0 3 0	**0 2 0**
12	0 1 2	
13	1 2 0	
14	2 1 0	
15	2 0 0	
16	**1 1 1**	
Итого	Достаточное значение параметра p=16	Достаточное значение параметра p=11

Таблица 2. Сравнение результатов алгоритмов на примере функции $y = 3x_3^3 + 2x_2^2 + 1x_1 + 2x_1x_2x_3 + 5$ (число наблюдений m=91)

№ п/п включения	Базовый алгоритм	Модификация 3
1	**0 0 3**	**0 0 3**
2	0 0 2	**1 1 1**
3	**0 0 1**	**0 2 0**
4	0 1 2	1 0 0
5	0 1 1	0 0 2
6	0 2 1	**0 0 1**
7	1 0 2	
8	**1 1 1**	
9	1 0 1	
10	2 0 1	
11	0 1 0	
12	**0 2 0**	
Итого	Достаточное значение параметра p=12	Достаточное значение параметра p=6

Еще разительнее оказались результаты сравнения алгоритмов при восстановлении функции $y = 3x_1^3 + 2x_2^2 + 5$ (таблица 3). В Варианте 2 в число перспективных «полезные» функции попадают самыми первыми ($p = 2$), а в базовом алгоритме обе войдут лишь при $p = 18$ (из 24 всего).

Таблица 3. Сравнение результатов алгоритмов на примере функции $y = 3x_1^3 + 2x_2^2 + 5$ (число наблюдений m=29)

№ п/п включения	Базовый алгоритм	Модификация 3
1	**3 0**	**3 0**
2	2 0	**0 2**
3	1 0	
4	2 1	
5	2 -1	
6	-1 0	
7	1 1	
8	1 -1	
9	1 2	
10	-2 0	
11	-1 -2	
12	-3 0	
13	-1 1	
14	-2 1	
15	1 -2	
16	0 3	
17	-2 -1	
18	**0 2**	
Итого	Достаточное значение параметра p=18	Достаточное значение параметра p=2

Приведем еще один пример сравнения алгоритмов для зависимости (1) $y = 3x_1^3 + 2x_2^2 + x_1^{-1}x_2^2 + 6x_1x_2 + 7x_2 + 5$. Здесь алгоритм Варианта 2, хотя и не намного, но также «опережает» базовый (Таблица 4).

Таким образом, появляются вполне определенные основания ожидать, что при использовании алгоритма этой модификации существенные для данной модели функции будут отбираться быстрее, нежели с помощью базового алгоритма. Как показывают проведенные эксперименты, последовательный отбор перспективных функций на основании их связи с остатками действительно оказывается более эффективным, чем принятый в базовом алгоритме принцип отбора перспективных функций. А отмеченная особенность предложенного алгоритма, т.е. проявленная им более высокая скорость выявления существенных функций, может, в конечном счете, отразится и на

минимально требуемом объеме информации, необходимом для восстановления адекватного описания модели.

Таблица 4. Сравнение результатов алгоритмов на примере функции $y = 3x_1^3 + 2x_2^2 + x_1^{-1}x_2^2 + 6x_1x_2 + 7x_2 + 5$ (число наблюдений m=52)

№ п/п включения	Базовый алгоритм	Модификация 3
1	**3 0**	**3 0**
2	2 0	**0 1**
3	1 0	-2 1
4	2 1	-1 1
5	**1 1**	1 2
6	1 2	**1 1**
7	-2 0	-3 0
8	-2 1	0 3
9	**0 2**	-2 0
10	0 3	-1 0
11	**0 1**	**-1 2**
12	-1 0	1 0
13	-3 0	**0 2**
14	-1 1	
15	**-1 2**	
Итого	Достаточное значение параметра p=15	Достаточное значение параметра p=13

Заключение

Основные результаты выполненной работы состоят в следующем. Предложен, программно реализован и численно апробирован ряд вариантов изменения базового алгоритма известного метода. Рассмотрены три варианта развития базового алгоритма метода Эглайса за счет изменений в ключевом этапе – в процедуре формирования набора перспективных функций. Наиболее удачным, в смысле скорости включения существенных функций в перспективные, оказалось применение «остаточного» признака. Согласно ему, очередным кандидатом в число перспективных является функция, имеющая наибольший по модулю коэффициент корреляции с остатками, представляющими собой погрешности аппроксимации, выполненной на базе функций, включенных, к этому моменту, в число перспективных.

Проведенные вычислительные эксперименты во множестве полиномиальных функций показали, что по сравнению с базовой версией алгоритма, такой принцип отбора позволяет выделять «полезные» базовые функции быстрее. В дальнейшем, представляется целесообразным обобщить полученные выводы и для класса трансцендентных функций.

Есть основания полагать, что реализация этого принципа поможет уменьшать задаваемый объем перспективного набора, что, в свою очередь, позволяет надеяться и на последующее ускорение процесса элиминации. То есть в базовом алгоритме присутствовало неявное, но достаточно жесткое требование к объему входных данных. В модифицированной версии это требование менее значимо. Благодаря внесенным изменениям, можно надеяться на более успешное включение всех полезных функций.

Отмечена особенность влияния объема исходной информации на скорость включения существенных функций в число перспективных. Есть основания полагать, что она определяется соотношением этого объема и числа входящих переменных.

Список литературы

1. Иващенко А.Б. Синтез аппроксимирующей функции при неизвестной структуре / А.Б. Иващенко, В.Н. Беловодский // Системный анализ и информационные технологии в науках о природе и обществе (САИТ-2011). – 2011. – Вып. 1. – С.157-166.

2. Эглайс В.О. Аппроксимация табличных данных многомерным уравнением регрессии / В.О. Эглайс // Вопросы динамики и прочности. – 1981. – Вып. 39. – С. 120 – 125.

Надійшла до редакції 25.09.2011.
Рецензент: канд. физ.-мат. наук, доц. Климко Г.Т.

А.Б. Іващенко, В.М. Бєловодський
Донецький національний технічний університет

Деякі варіації методу Еглайса синтезу апроксимуючих функцій. Робота присвячена обговоренню низки варіантів зміни базового алгоритму методу синтезу апроксимуючих функцій, запропонованого В. Еглайсом в кінці 70-х років минулого століття. За допомогою програм, що реалізують запропоновані модифікації, і серії обчислювальних експериментів проведено порівняльний аналіз результатів використання модифікованих алгоритмів з початковим.
Ключові слова: апроксимація, модифікація, базовий алгоритм, регресійна модель, залежність, банк функцій, елімінація, кореляція залишків з відгуком.

A.B. Ivashchenko, V.N. Belovodskiy
Donetsk National Technical University

Some Variations of the Eglais's Method of Synthesis of Approximating Functions. The work is devoted to discussion of a number of variants for changing the base algorithm of the method of synthesis of approximating functions, proposed by V. Eglais at the end of 70th of the last century. With the help of programs, that implement the proposed modifications, and a series of numerical experiments a comparative analysis of the results of exploitation of modified algorithms with original one has been made.
Keywords: approximation, modification, base algorithm, regressive model, dependence, bank of functions, elimination, correlation of residuals with response.