



ISSN: 2658–5782

Номер 2

2021

МНОГОФАЗНЫЕ СИСТЕМЫ

mfs.uimech.org





Использование свободных программ для обработки и визуализации результатов научных исследований¹

Насибуллаев И.Ш.

Институт механики им. Р.Р. Мавлютова УФИЦ РАН, Уфа

В работе представлены обзор и примеры применения свободного программного обеспечения для обработки, анализа и визуализации результатов вычислительных и натуральных экспериментов. Рассмотрена реализация двух стандартных методов аппроксимации и анализа данных: метод наименьших квадратов и ортогональное центральное композиционное планирование. Первый метод позволяет провести аппроксимацию табличных данных в виде полинома произвольной степени. При повышении степени полинома точность аппроксимации возрастает, но, из-за роста числа обусловленности матрицы метода, корректность аппроксимации снижается. Показано, что нормировка исходных данных позволяет повысить максимальный порядок полинома, при котором не будет потери точности из-за плохо обусловленной матрицы. Предложена реализация метода на языке программирования Scilab. Второй метод позволяет получить аппроксимацию второго порядка для произвольного количества переменных, оценить линейные и квадратичные вклады отдельных переменных и их парных взаимодействий. Представлено описание метода и предложена схема построения расширенной матрицы планирования эксперимента. Метод реализован на языке программирования C++, а для определения погрешности полученной аппроксимации предлагается скрипт, написанный на языке Scilab. Для получения более точной аппроксимации можно подобрать функцию, учитывающую асимптотические особенности результатов вычислительного эксперимента. Пример такой аппроксимации реализован в программе символьных вычислений wxMaxima. Представленные программные реализации методов аппроксимации дополнительно создают код в формате Gnuplot и Scilab, что повышает эффективность для последующей визуализации или анализа аппроксимаций. Приведены примеры визуализации исследования в \LaTeX -пакете PGFPLOTS и в программе Gnuplot с использованием терминала epslatex. Оба подхода удобны при оформлении результатов исследования в настольной издательской системе \LaTeX .

Ключевые слова: свободное программное обеспечение, метод наименьших квадратов, ортогональный центральный композиционный план, Scilab, wxMaxima, Latex, Gnuplot

1. Введение

В последние годы свободное программное обеспечение (ПО) активно развивается и по широкому спектру задач способно заменить коммерческое. Например, многие численные научные и инженерные задачи, решаемые в проприетарном пакете прикладных программ *MATLAB* [1], можно реализовать в свободном ПО *Scilab* [2, 3], GNU Octave [4, 5] или *Maxima* [6, 7]. Альтернативой для выполнения символьных и численных вычислений

в проприетарных системах компьютерной алгебры *Mathematica* [8] и *Maple* [9] является свободное ПО *Maxima* с графическим интерфейсом *wxMaxima* [10]. В работе [11] приведен обзор свободного ПО для численного моделирования инженерных задач методом конечных элементов.

При моделировании физических процессов (течение жидкости или газа, упругие или пластические деформации твердого тела) необходимо учитывать зависимость материальных параметров (плотность, вязкость, теплоемкость и т.д.) от факторов внешней среды (температура, давление и т.д.). Обычно эти зависимости представлены в виде табличных данных (результатов натурального эксперимента). Для решения физических уравнений зави-

¹Работа выполнена за счет средств государственного задания № 0246-2018-007.

симости аппроксимируют в виде аналитических непрерывных функций (кусочно-линейная, полиномиальная). Одним из методов аппроксимации является метод наименьших квадратов [12] (МНК, least square method, LSM, polynomial regression), который позволяет построить аналитическую функцию заданного вида с наименьшей невязкой относительно табличных данных.

На первом этапе исследования проводится планирование эксперимента (design of experiments, DoE) [13], натурального или вычислительного, с определением входных параметров (переменные, факторы, variables), количеством уровней для каждого из них и выбирается диапазон значений факторов.

На следующем этапе составляется матрица эксперимента (экспериментальный план), содержащая упорядоченные наборы факторов [14]. По данной матрице проводится серия экспериментов и составляется приближенная аналитическая формула для целевой функции (response function). Из анализа аппроксимации и исходных данных определяются: значимость факторов и их взаимосвязь; адекватность выбранного плана эксперимента и набора факторов; наличие скрытых переменных, влияющих на результат эксперимента; влияние на целевую функцию отсроченных эффектов значимых факторов; достаточность экспериментов для обобщения результатов; оценка систематических и случайных погрешностей; влияние внешних условий на повторяемость эксперимента.

На завершающем этапе проводится подготовка презентации или рукописи. При подготовке публикации результатов широко используется свободная настольная издательская система \LaTeX [15–17], которая поддерживает как подготовку текста с использованием принятого в издательстве шаблона оформления, так и включения в него математических формул и графических объектов. Многие издательства во всем мире рассматривают издательскую систему \LaTeX как альтернативу текстовому процессору *MS Word* [18] (в ряде журналов публикации принимаются либо только в формате \LaTeX , либо формат \LaTeX является предпочтительным). Для визуализации результатов исследования (построение графиков, схем и диаграмм) удобно использовать стандартные \LaTeX -пакеты: для построения графиков — *PGFPlots* [19]; для схем и диаграмм — *TikZ* [20], *PSTricks* [21] или внешние свободные программы, например, *Gnuplot* [22] (альтернатива коммерческому ПО *Origin* [23]). Пакет *listings* [24] позволяет отображать исходный код программ и тестов, записанных с помощью языков разметки, используя форматирование и подсветку синтаксиса. Пакет *beamer* [25] предоставляет свободную аль-

тернативу для коммерческой программы создания презентаций *MS PowerPoint* [18].

В настоящей работе рассматриваются два способа обработки результатов научного исследования: метод наименьших квадратов с предварительной нормировкой исходных данных для повышения точности аппроксимации и ортогональное центральное композиционное планирование (ОЦКП, orthogonal central composite design, OCCD), позволяющее определить вклад факторов в целевую функцию и их взаимосвязь. Первый метод реализован в ПО *Scilab*, а второй написан на языке программирования C++ с улучшением точности аппроксимации по асимптотическому поведению искомой функции с реализацией в ПО *wxMaxima*. Исходные коды программ приведены с использованием пакета *listings*. Для построения схем использовался пакет *TikZ*. Пример построения графиков с помощью пакета *PGFPlots* приведен в Приложении А, а с помощью пакета *Gnuplot* — в тексте статьи и в Приложении В.

2. Реализация метода наименьших квадратов в Scilab

Рассмотрим способ построения аппроксимации экспериментальных данных y_i ($i = (1, m)$, m — количество опытов) в виде полинома n степени с помощью МНК. Аппроксимирующая функция будет нелинейной, но сам метод остается линейным — коэффициенты аппроксимации b_j ($j = (0, n)$) умножаются на степени x^j , которые рассматриваются как независимые переменные. Сам полином имеет вид:

$$f(x, b) = \sum_{j=0}^n b_j x^j. \quad (1)$$

Сумма квадратов отклонений

$$S = \sum_{i=1}^m (y_i - f(x, b))^2 = \sum_{i=1}^m \left(y_i - \sum_{j=0}^n b_j x_i^j \right)^2$$

будет минимальной, если $\partial S / \partial b_k = 0$ для каждого $k = (0, n)$, что приводит к системе $n + 1$ линейных алгебраических уравнений (СЛАУ, system of linear equations, SLE) с неизвестными переменными b_j :

$$\sum_{i=1}^m \sum_{j=0}^n b_j x_i^{j+k} = \sum_{i=1}^m y_i x_i^k, \quad k = (0, n),$$

или в матричной форме

$$Ab = B, \quad A_{k,j} = \sum_{i=1}^m x_i^{j+k}, \quad B_k = \sum_{i=1}^m y_i x_i^k. \quad (2)$$

Численная реализация алгоритма имеет следующие особенности: при больших значениях x_i максимальный элемент в матрице A (2) равен x_i^{2n} , а минимальный — m и отношение этих элементов x_i^{2n}/m будет значительным. Например, для $m = 10$, $n = 5$ и $x = 100$, получим значение 10^{19} — матрица A будет плохо обусловлена и результат аппроксимации в виде полинома высокого порядка ($n > 3$) будет некорректен. Для ограничения максимального элемента матрицы A используем подход, предложенный в [26]: проведем нормировку диапазона экспериментальных данных и их значений:

$$\begin{aligned} x_i &\leftarrow x_i/s_x, y_i \leftarrow y_i/s_y, \\ s_x &= \max(x), s_y = \max(y). \end{aligned} \quad (3)$$

Для оценки качества матрицы A определяется число обусловленности cn , а точность аппроксимации рассчитывается по среднему квадратичному отклонению полинома от экспериментальных данных σ :

$$\sigma = \sqrt{\frac{1}{n} \sum_1^m (y_i - f(x_i, b))^2}. \quad (4)$$

Запишем на языке программирования *Scilab* функцию определения коэффициентов аппроксимации b_j (1), среднего квадратичного отклонения σ и числа обусловленности sn с нормировкой экспериментальных данных (3) ($d \neq 0$ — данные нормируются, $d = 0$ — не нормируются):

```
function [b, sigma, cn] = lsrp(x,y,m,d)
if d then
sx = max(x); sy = max(y);
else
sx=1.0; sy=1.0;
end
xn = x / sx;
for i=1:m+1
B(i) = y * (xn.^(i-1))' / sy;
for j=1:m+1
A(i,j) = sum(xn.^(i+j-2));
end;
end;
cn = cond(A);
b = inv(A)*B * sy;
for j=1:m+1
b(j) = b(j) / sx.^(j-1);
end;
ya=0;
for j=0:m
ya = ya + b(j+1) * x.^j;
end
sigma = sqrt((ya - y) * (ya - y)'/length(x));
endfunction
```

Для визуализации результатов напишем функцию, рассчитывающую $(x, f(x, b))$ (1) на диапазоне экспериментальных данных (x_1, x_m) , но для заданного числа точек p (при увеличении p график будет более гладким):

```
function [xf,yf] = lsrpFine(x,b,n,p)
xf = linspace(min(x),max(x), p);
Xf = zeros(p,n+1);
for i = 1:n+1
Xf(:,i) = xf(:).^i-1;
end
yf = (Xf*b)';
endfunction
```

Проведем аппроксимацию температурной зависимости изобарной теплоемкости воды $c_p(T)$ [27]. В *Scilab* графики строятся командой *plot* с парами векторов, содержащих координаты точек x и y , где опционально можно задать формат стиля линий. Форматный вывод в консоль значений переменных оформляется командой *printf*. Запись значений переменных в файл осуществляется командой *write*. В коде программы добавим вывод в консоль формул полиномов с рассчитанными коэффициентам в формате *Gnuplot* — это позволит сэкономить время при подготовке графика в *Gnuplot*, — а также выведем в файлы с именами fn (n — порядок полинома, $path$ — строка, содержащая путь к рабочему каталогу) результаты аппроксимации в виде столбцов $(x, f(x, b))$.

```
x=[0,10,20,30,40,50,60,70,80,90,100];
y=[4.2176,4.1921,4.1818,4.1784,4.1785,4.1806,
4.1843,4.1895,4.1963,4.2050,4.2159]*1e3;
k=5; p=101; path = "";

for n=1:k
ya = zeros(p); b = zeros(n);
[b,sigman,cn] = lsrp(x,y,n,1);
[xa,ya] = lsrpFine(x,b,n,p)
plot(x,y,"o",xa,ya);
printf("i = %2d: sigma = %10.4e,
cn = %10.4e, b = ", n, sigman, cn);
printf("%10.4e ", b);
printf("\n f%d(x)=%9.4e", n,b(1));
for i=1:n
printf("%+9.4e*x**%d", b(i+1),i);
end
printf("\n");
write("f"+string(n), [xa' ya']);
end
write("f0", [x' y']);
```

Запускаем *Scilab*, помещаем все три фрагмента кода в *SciNote* (в меню выбираем *Applications*→*SciNotes*), сохраняем (комбинация клавиш *Ctrl+s*), запускаем (клавиша *F5*). Результаты выполнения кода будут выведены в консоль программы. Изменяя параметр p проведем расчет с нормировкой экспериментальных данных и без нее. В первом случае максимальное число обусловленности cn будет на 12 порядков меньше, чем во втором, т.е. в приведенном примере нормировка позволяет сохранить в результатах расчетов на 12 значащих цифр больше. Отметим, что на cn влияет только нормировка диапазона данных (x_1, x_m) .

Используем созданный программой файл с экспериментальными данными f_0 и выражения полиномов, выведенных в консоль программы, для создания более качественного графика в *Gnuplot*. При построении графика из файла данных в команде *plot* указываются имя файла и через двоеточие номера столбцов для оси абсцисс и ординат в виде 1 : 2 (в 3D добавляется аппликата, например, 1 : 2 : 3). Если данные нужно преобразовать, например, провести масштабирование с коэффициентом s и смещение на величину d , то формула записывается в круглых скобках с префиксом номера столбца в виде символа $\$$, например 1 : 2 заменяется на $(\$1 * s + d) : 2$. В примере единицы из Дж переведены в кДж с соответствующим масштабированием табличных данных и аппроксимаций на величину $sy = 10^{-3}$. Формулу для полинома добавляем командой *set label*.

```
set terminal epslatex size 10cm,6cm\
color colortext
set decimalsign ', '
set border lw 2
set style line 1 lc rgb 'black' ps 1.5 pt 6
set style line 2 lc rgb 'red' lt 1 dt 3 lw 5
set style line 3 lc rgb 'orange' lt 1 dt 2 lw 5
set style line 4 lc rgb 'cyan' lt 1 dt 1 lw 5
set style line 5 lc rgb 'blue' lt 1 dt 1 lw 5
set style line 6 lc rgb 'green' lt 1 dt 1 lw 5
set key at 50,4.218 spacing 0.9 box dt 3
set output 'epslatex.tex'
f1(x)=4.1875e+3+0.10418*x
f2(x)=4.2090e+3-1.3275*x+1.4317e-2*x**2
f3(x)=4.2145e+3-2.1985*x+3.7158e-2*x**2-\
1.5227e-4*x**3
f4(x)=4.2169e+3-3.0486*x+7.9662e-2*x**2-\
8.3234e-4*x**3+3.4003e-6*x**4
f5(x)=4.2175e+3-3.4918*x+1.1618e-1*x**2-\
1.8610e-3*x**3+1.5179e-5*x**4-4.7115e-8*x**5
set xlabel '$T$, $^\circ$C'
set ylabel '$c_p(T)$, \dfrac{\text{кДж}}{\text{кг}\cdot\text{К}}$'
set label 1 at 53,4.212 \
'$c_p(T) = \sum\limits_{i=0}^n b_i T^i$'
sy=1.e-3
plot "f0" u 1:($2*sy) title 'data' w p ls 1,\
f1(x)*sy title '$n=1$' w l ls 2,\
f2(x)*sy title '$n=2$' w l ls 3,\
f3(x)*sy title '$n=3$' w l ls 4,\
f4(x)*sy title '$n=4$' w l ls 5,\
f5(x)*sy title '$n=5$' w l ls 6
```

График показан на рис. 1. Видно, что качественная аппроксимация достигается при $n = 4$ и $n = 5$ с нормированием исходных данных.

3. Ортогональное центральное позиционное планирование в C++

При проведении натуральных или вычислительных экспериментов при варьировании нескольких параметров необходим инструмент, позволяющий

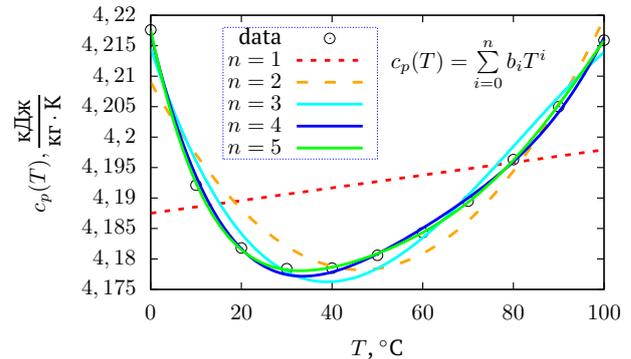


Рис. 1. Результаты аппроксимации МНК в *Scilab* с визуализацией в *Gnuplot*. Символы – экспериментальные данные [27], линии – аппроксимация (1)

определить вклад каждого из параметров в измеряемую величину. Одним из таких инструментов является полный факторный эксперимент (ПФЭ, factorial design) [14, 28], аппроксимирующий измеряемую величину (целевую функцию) с помощью полинома первой степени, учитывающего линейный вклад параметров (факторов) и их взаимодействий друг с другом. Величина факторов берется на двух уровнях (нижний и верхний), а целевая функция Y определяется для всех возможных комбинаций значений факторов. Для n факторов X_i , где $i = (1, n)$, необходимо провести $n_f = 2^n$ опытов. Для учета квадратичных вкладов факторов ПФЭ проводится на трех уровнях, что требует проведения 3^n опытов. Метод ОЦКП [29, 30] позволяет учесть квадратичные факторы меньшим, чем в ПФЭ, количеством опытов. Приведем схему параметрического анализа данных методом ОЦКП с учетом только парных взаимодействий. Для естественного значения i -го фактора X_i обозначим значение на нижнем и верхнем уровнях через $X_{i,-1}$ и $X_{i,+1}$, определим центральное значение $X_{i,c}$ и интервал $X_{i,\ell}$:

$$X_{i,c} = \frac{X_{i,-1} + X_{i,+1}}{2}, \quad X_{i,\ell} = \frac{X_{i,+1} - X_{i,-1}}{2}. \quad (5)$$

Кодированным значением i -го фактора x_n является линейное аффинное преобразование интервала $[X_{i,-1}, X_{i,+1}]$ в интервал $[-1, +1]$ по формуле

$$x_i = \frac{X_i - X_{i,c}}{X_{i,\ell}}. \quad (6)$$

Естественные значения фактора можно восстановить по кодированному используя формулу

$$X_i = X_{i,\ell} x_i + X_{i,c}. \quad (7)$$

Общее количество опытов n_t состоит из n_f опытов ПФЭ, n_* опытов в «звездных» точках и одного опыта в центральной точке:

$$n_t = n_f + n_* + 1, n_f = 2^n, n_* = 2n. \quad (8)$$

Вводится фиктивный фактор $x_0 = 1$ и целевая функция аппроксимируется полиномом вида:

$$f_k(x_1, \dots, x_n) = b_0 + \sum_{i=1}^n b_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n b_{ij} x_i x_j + \sum_{i=1}^n b_{ii} (x_i^2 - \beta), \quad (9)$$

где $k = (1, m)$, m – количество целевых функций; параметр $\beta = \sqrt{n_f/n_t}$; двойная сумма содержит n_c слагаемых:

$$n_c = \frac{n(n-1)}{2!}. \quad (10)$$

Для кодированных значений факторов строится расширенная матрица вычислительного эксперимента содержащая следующие строки (см. рис. 2):

- Матрица ПФЭ A_1 (размер матрицы $n_f \times n$), содержащая все возможные комбинации нижнего и верхнего уровней $x_i = \pm 1$.
- «Звездные» точки A_2 ($n_* \times n$), в которых $x_i = 0$ для всех i кроме $i = k$, для которой $x_k = \pm \alpha$, где α – расстояние до центральной точки плана

$$\alpha = \sqrt{\frac{1}{2}(\sqrt{n_f n_t} - n_f)}.$$

Для $n > 2$ величина $\alpha > 1$, что накладывает ограничения при выборе диапазона естественных значений факторов. Диапазон не должен

противоречить физическому смыслу фактора. Например, нижний уровень для пройденного пути не может быть отрицательным, а верхний уровень для температуры воды не должен превышать 100°C .

- Центральная точка A_3 ($1 \times n$), в которой все значения $x_i = 0$.

После этого добавляются столбцы (см. рис. 2):

- слева добавляется столбец A_0 (размер матрицы $n_t \times 1$) с фиктивным параметром $x_0 = 1$;
- справа A_4 – кодированные значения для парных взаимодействий $x_i x_j$ ($n_t \times n_c$);
- справа A_5 – квадратичные слагаемые ($n_t \times n$);
- для удобства добавляется n столбцов A_6 с естественными значениями факторов X_i и n_y столбцов A_7 для значений целевых функций;
- столбцы A_8 с величиной погрешности аппроксимации:

$$\varepsilon = \left| 1 - \frac{f_k(x_1, \dots, x_n)}{Y_k(x_1, \dots, x_n)} \right| \cdot 100\%. \quad (11)$$

Коэффициенты полинома (9) определяют для каждой целевой функции Y_k отдельно по формуле:

$$b_j = \frac{\sum_{i=0}^{2n+n_c} Y_{k,i} x_{j,i}}{\sum_{i=0}^{n_t} x_{j,i}}, \quad (12)$$

где j и i – индексы столбца и строки расширенной матрицы эксперимента.

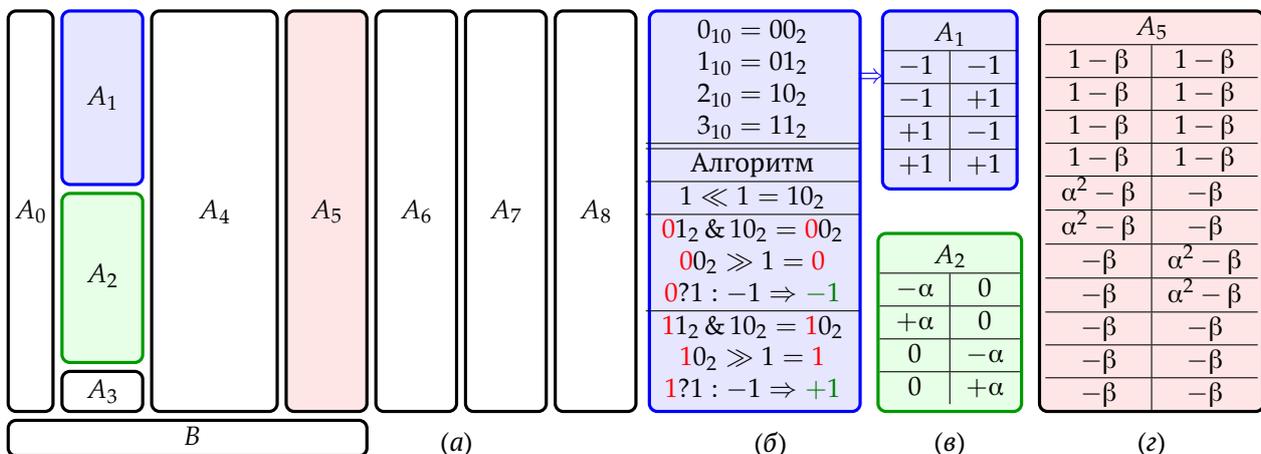


Рис. 2. Структура расширенной матрицы ОЦКП (а) и структура матриц A_1 (б); A_2 (в); A_5 (з). Красные цифры – определяемый бит для опытов под номерами 1 и 3; зеленые числа – кодированное значение фактора x_1

Таблица 1. Расширенная матрица вычислительного эксперимента, коэффициенты и относительные погрешности полинома (9)

x_0	x_1	x_2	x_1x_2	$x_1^2 - \beta$	$x_2^2 - \beta$	X_1	X_2	Y	$\varepsilon, \%$
1	-1	-1	1	0.3333	0.3333	1	0.5	12.45	1.24
1	1	-1	-1	0.3333	0.3333	5	0.5	7.438	0.631
1	-1	1	-1	0.3333	0.3333	1	2.5	9.217	0.178
1	1	1	1	0.3333	0.3333	5	2.5	2.755	4.54
1	-1	0	0	0.3333	-0.6667	1	1.5	9.545	1.45
1	1	0	0	0.3333	-0.6667	5	1.5	3.481	2.24
1	0	-1	0	-0.6667	0.3333	3	0.5	8.219	2.46
1	0	1	0	-0.6667	0.3333	3	2.5	3.744	3.77
1	0	0	0	-0.6667	-0.6667	3	1.5	4.394	1.39
b_0	b_1	b_2	$b_{1,2}$	$b_{1,1}$	$b_{2,2}$				
6.804	-2.924	-2.066	-0.364	2.028	1.496				

Приведем пример обработки результатов вычислительного эксперимента по данным, полученным при моделировании системы жидкостного охлаждения микрозахвата [31] для $n = 2$ (скорость охлаждающей жидкости $X_1 = [1, 5]$, коэффициент теплопередачи $X_2 = [0.5, 1.5]$) с одной целевой функцией Y (температура нагрева радиатора). Порядок построения расширенной матрицы вычислительного эксперимента следующий:

- задаем диапазоны естественных значений факторов X_i ;
- вычисляем кодированные значения факторов x_i (матрицы A_0, \dots, A_5);
- по кодированным значениям факторов x_i вычисляем матрицу A_6 естественных значений факторов X_i по формуле (7);
- для значений из матрицы A_6 проводим серию вычислительных экспериментов и заполняем матрицу A_7 ;
- вычисляем коэффициенты полинома (9) по формуле (12) и определяем относительные погрешности (11).

Для рассматриваемого набора данных расширенная матрица, коэффициенты аппроксимации и относительные погрешности целевой функции приведены в табл. 1.

В программе сначала опишем переменные (8) и (10), определим количество целевых функций n_y , их значения a для n_t опытов и диапазон данных естественных переменных X :

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;
int main() {
    int n = 2, nstar = 2*n, nf = pow(2,n), ny = 1;
    int nc = (n-1)*n/2, nt = nf+nstar+1;
    double alpha = sqrt(0.5*(sqrt((double)nf*nt)-
        nf)), beta = sqrt((double)nf/nt);
    double a[nt+1+ny][nf+nc+n+ny+1]={0};
    double x[2][n] = {{1,0.5},{5,2.5}};
    double y[ny][nt] = {{12.448999, 7.438013,
        9.217004, 2.749994, 9.545007, 3.481012,
        8.218988, 3.744012, 4.394006}};
    int i1, i2, i3, i4, i5;
```

Здесь элементы расширенной матрицы a задаются равными нулю, поэтому нулевые элементы «звездных» точек A_2 и центральная точка A_3 определены по умолчанию (рис. 2). Определим матрицу A_0 фиктивного фактора x_0 (рис. 2(a)):

```
for (i1 = 0; i1 < nt; i1++) a[i1][0] = 1.0;
```

При построении матрицы ПФЭ представим номер опыта (нумерацию будет вести с нулевого номера) в двоичном представлении с количеством разрядов равным n (рис. 2(б)). Для n_f опытов получим таблицу, в которой номер столбца соответствует номеру опыта, а столбец (разряд числа) дает значение кодированного фактора в виде «0» для нижнего уровня и «1» — для верхнего. Текущая позиция разряда i (номер позиции определяется справа налево с крайним правым номером равным 0) находится левым логическим битовым сдвигом единицы ($1 \ll i$); побитовая операция «логическое И» (&) для i и номера опыта позволяет определить значение бита в i -м разряде; правый логический битовый сдвиг (\gg) операции «И» на i разрядов перемещает бит с i -й позиции на 1-ю (результатом будет «0» или «1»); с помощью тернарной опера-

ции («?») заменяем «0» на «-1». После применения данного алгоритма для n_f опытов получим матрицу ПФЭ A_1 :

```
for (i1 = 0; i1 < nf; i1++)
  for (i2 = 0; i2 < n; i2++)
    a[i1][i2+1] = (i1 & (1 << i2)) >> i2 ? 1:-1;
```

Отметим, что для ПФЭ с тремя уровнями $\{-1, 0, +1\}$ предложенный алгоритм также применим, только использовать нужно не двоичную запись числа, а троичную и вместо тернарной операции нужно из результата вычесть 1.

Задаем кодированные значения «звездных» точек A_2 (рис. 2(в)):

```
for (i1 = 0; i1 < n; i1++)
  for (i2 = 0; i2 < n; i2++) {
    a[2*i1+nf][i2+1] = i1 == i2 ? -alpha : 0.0;
    a[2*i1+nf+1][i2+1] = i1 == i2 ? alpha : 0.0;
  }
```

Определяем матрицу парных взаимодействий A_4 :

```
for (i1 = 0; i1 < nf; i1++) {
  i4=n;
  for (i2 = 1; i2 < n; i2++)
    for (i3 = i2+1; i3 <= n; i3++)
      a[i1][++i4] = a[i1][i2]*a[i1][i3];
}
```

и матрицу квадратичных факторов A_5 (рис. 2(г)):

```
for (i1 = 0; i1 < nt; i1++)
  for (i2 = 1; i2 <= n; i2++)
    a[i1][n+nc+i2] = pow(a[i1][i2], 2) - beta;
```

Матрица A_6 строится по формуле (7):

```
for (i1 = 0; i1 < nt; i1++)
  for (i2 = 0; i2 < n; i2++)
    a[i1][2*n+nc+i2+1] = 0.5 * (a[i1][i2+1] *
      ( x[1][i2]-x[0][i2] ) + x[0][i2]+x[1][i2] );
```

Матрица A_7 содержит результаты экспериментов:

```
for (i1 = 0; i1 < nt; i1++)
  for (i2 = 0; i2 < ny; i2++)
    a[i1][3*n+nc+1+i2] = y[i2][i1];
```

Определяем сумму в знаменателе формулы (12):

```
for (i1 = 0; i1 <= 2*n+nc; i1++)
  for (i2 = 0; i2 < nt; i2++)
    a[nt][i1] += pow(a[i2][i1], 2);
```

и находим коэффициенты полинома (12) (значения b_j приведены в табл. 1):

```
for (i1 = 0; i1 < ny; i1++)
  for (i2 = 0; i2 <= 2*n+nc; i2++)
    for (i3 = 0; i3 < nt; i3++)
      a[nt+1+i1][i2] +=
        y[i1][i3]*a[i3][i2]/a[nt][i2];
```

Относительная погрешность аппроксимации A_8 согласно (11):

```
for (i1 = 0; i1 < nt; i1++)
  for (i3 = 1; i3 <= ny; i3++) {
    for (i2 = 0; i2 <= 2*n+nc; i2++)
      a[i1][3*n+nc+ny+i3]+=a[nt+i3][i2]*a[i1][i2];
    a[i1][3*n+nc+ny+i3]=100.0*abs(1.0-
      a[i1][3*n+nc+ny+i3]/a[i1][3*n+nc+i3]);
  }
```

Выводим в консоль расширенную матрицу планирования эксперимента (табл. 1):

```
cout.precision(4);
for (i1 = 0; i1 < nt+1+ny; i1++) {
  for (i2 = 0; i2 <= 3*n + nc+ny; i2++)
    cout << setw(7) << a[i1][i2] << " ";
  cout << endl;
}
```

Выводим в консоль запись полинома (9) в формате функции *Gnuplot* для $p = 0$ или *Scilab* для $p = 1$:

```
int p=0; // 0-Gnuplot, 1-Scilab
cout.precision(8);
cout << "b=" << beta <<endl << "s=1.0" <<endl;
for (i1 = 0; i1 < ny; i1++) {
  i2=nt+1+i1;
  cout << (p?"function f=":"")<<"f"<<i1<<"(x1";
  for (i3 = 2; i3 <= n; i3++) cout << ",x"<<i3;
  cout << ")"<<(p?"\n f=":"="("<<a[i2][0];
  for (i3 = 1; i3 <= n; i3++)
    cout<<showpos<<a[i2][i3]<<"x"<<noshowpos<<i3;
  i5 = n;
  for (i3 = 1; i3 < n; i3++)
    for (i4 = i3+1; i4 <= n; i4++)
      cout << showpos << a[i2][++i5]<<"*x"<<
        noshowpos<<i3<<(p?".":"")<<"*x"<<i4;
    for (i3 = 1; i3 <= n; i3++)
      cout << showpos << a[i2][++i5] <<"*(x"
        << noshowpos<<i3<<"**2-b)";
    cout << (p?"\nendfunction":"*s") << endl;
  }
  return 0;
}
```

Gnuplot скрипт для построения полученной аппроксимации (рис. 3):

```
set terminal epslatex size 10cm,12cm\
  color colortext
set decimalsign '.'
set border lw 2
set grid lw 0.1 dt 1 lc rgb "#AAAAAA"
set key off
T0=293.15
b=0.66666667
s=1.0
f0(x1,x2)=(6.804115-2.9236652*x1-2.0658317*x2-\
  0.364006*x1*x2+2.0276695*(x1**2-b)+\
  1.49616*(x2**2-b))*s
set output "occd.tex"
set multiplot layout 2,1 rowsfirst
set xlabel '$X_1$' offset 0,0.5
set ylabel '$Y_1$' offset 1,0
plot "um" using 1:($7-T0) w p pt 7 ps 2,\
  f0((x-3)/2,0) w l lw 4 dt 1
```

```
set xlabel '$X_2$' offset 0,0.5
set ylabel '$Y_1$' offset 1,0
plot "alpha" using 1:($7-T0) w p pt 7 ps 2,\
f0(0,(x-1.5)/1) w l lw 4 dt 1
unset multiplot
```

Здесь стоит обратить внимание на то, что при вызове функции $f0$ естественные значения факторов кодируются согласно (6).

Точность аппроксимации определяется по формуле (11). Приведем пример вычисления погрешностей с помощью Scilab скрипта:

```
function f=f0(x1,x2)
f=6.804115-2.9236652*x1-2.0658317*x2-...
0.364006*x1.*x2+2.0276695*(x1^2-b)+...
1.49616*(x2^2-b)
endfunction
x1 = [1,5,1,5,1,5,3,3,3];
x2 = [0.5,0.5,2.5,2.5,1.5,1.5,0.5,2.5,1.5];
y1 = [12.45,7.438,9.217,2.75,9.545,3.481,...
8.219,3.744,4.394];
x1c=0.5*(x1(1)+x1(2));x1l=0.5*(x1(2)-x1(1));
x2c=0.5*(x2(1)+x2(3));x2l=0.5*(x2(3)-x2(1));
b = 2.0/3.0;
eps = [];
epsmax = [];
epsav = [];
eps = abs(1.0-f0((x1-x1c)/x1l,(x2-x2c)/x2l)...
./ y1)*100.0;
epsmax = max(eps);
epsav = sum(eps)/9.0;
printf('%f %f %f %f %f %f %f %f %f\n', eps);
printf('%f %f\n', epsmax, epsav);
```

В скрипте отметим следующее: «...» означает продолжение выражения на следующей строке; операция «.*» — поэлементное умножение векторов; «^» — возведение в степень; центр интервала естественного значения фактора и его плечо определяется по формуле (5), а в аппроксимации используются закодированные значения (6); переменные eps , $epsmax$ и $epsav$ содержат вектор относительных погрешностей, максимальную и среднюю погрешности, соответственно. Матрица относительных погрешностей A_8 приведена в табл. 1.

Проведем анализ полученных данных. Значение целевой функции Y_m в центральной точке вычисляется по формуле

$$Y_m = b_0 - \beta(b_{1,1} + b_{2,2}) \approx 4.455.$$

Ведущими линейными и квадратичными факторами являются x_1 и x_2 , т.к. $|b_1| > \chi$, $|b_2| > \chi$, $|b_{1,1}| > \chi$, $|b_{2,2}| > \chi$, где χ — критерий значимости фактора. Для определенности выберем

$$\chi = 0.05 * (|b_1| + |b_2| + |b_{1,2}| + |b_{1,1}| + |b_{2,2}|) \approx 0.44,$$

т. е. отдельное слагаемое значимо, если его вклад превышает 5%. Парное взаимодействие не оказывает значительного влияния $|b_{1,2}| < \chi$.

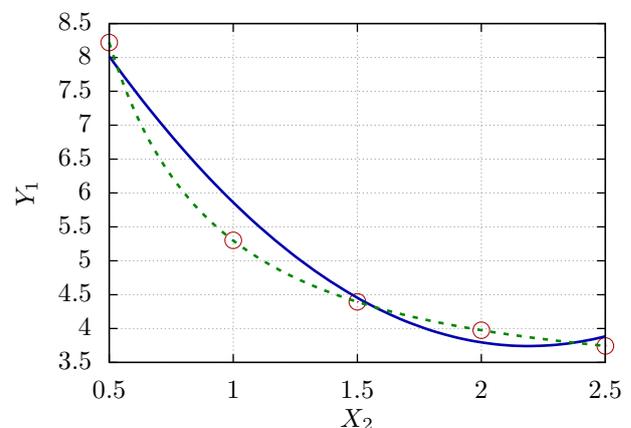
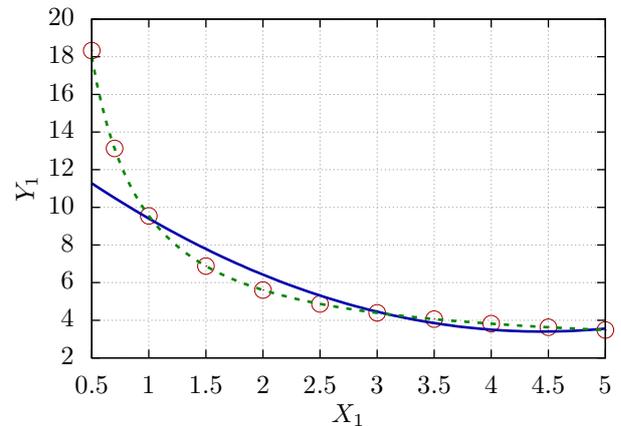


Рис. 3. Зависимость целевой функции от факторов X_1 ($x_2 = 0$, сверху) и X_2 ($x_1 = 0$, снизу). Символы — данные вычислительного эксперимента [26, 31], сплошные линии — аппроксимация ОЦКП (9); пунктирные линии — асимптотическая интерполяция (13)

При обработке данных натуральных экспериментов значимость фактора определяется с помощью распределения Стьюдента, а адекватность аппроксимации ОЦКП определяется по критерию Фишера (подробное изложение критериев приведено в работе [32]).

При увеличении значений x_1 и x_2 значение целевой функции линейно уменьшается ($b_1 < 0$, $b_2 < 0$), а для $x_1 > -0.5$ и $x_2 > -0.5$ это уменьшение компенсируется квадратичными слагаемыми (коэффициенты при линейных и квадратичных слагаемых имеют противоположные знаки).

Аппроксимация в точках ОЦКП (табл. 1) имеет небольшую относительную погрешность относительно экспериментальных данных внутри заданного диапазона данных. Из рис. 3 видно, что вдоль центральных линий, проходящих через три точки (центральная $X_1 = 3$ и $X_2 = 1.5$ и две «звездные» $X_1 = \{1, 5\}$ и $X_2 = \{0.5, 2.5\}$), погрешность в этих точках минимальна (на рис. 4 центральные линии

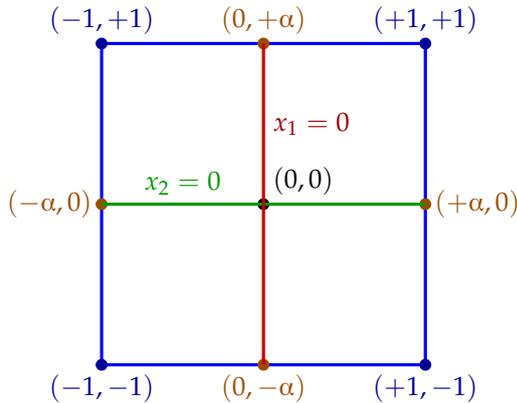


Рис. 4. Положение факторов по расширенной матрице ОЦКП для $n = 2$ и центральные линии $x_1 = 0$ и $x_2 = 0$

в кодированных значениях обозначены $x_1 = 0$ и $x_2 = 0$). В промежуточной области погрешность возрастает. За редким случаем (когда математическая модель исследуемого объекта полностью совпадает с используемым полиномом) аппроксимация не допускает экстраполяцию за пределы заданного диапазона — чем дальше находится точка вне диапазона данных, тем выше погрешность и аппроксимация становится неадекватной экспериментальным или расчетным данным. Например, для $X_1 = [0.5, 1]$ погрешность составляет 41%. Ширина диапазона данных также имеет значение: для $X_1 = [4.5, 5]$ и $X_2 = [2, 2.5]$ характер функции качественно меняется — вместо насыщения функция переходит к росту. Для того чтобы получить достаточно точную и адекватную аппроксимацию рекомендуется широкий диапазон значений факторов разбивать на несколько частей и строить ОЦКП для каждой из них.

4. Пакет символьных вычислений wxMaxima

Рассмотрим альтернативную, предложенную в предыдущем разделе, аппроксимацию с помощью интерполяционной функции. Используемые данные имеют следующий физический смысл: факторы X_1 и X_2 — скорость течения охлаждающей радиатор жидкости и коэффициент теплопередачи между жидкостью и радиатором, а целевая функция Y_1 является температурой нагрева радиатора относительно температуры окружающей среды. С учетом слабого парного взаимодействия $b_{1,3}$, а также асимптотических особенностей ($Y_1 \rightarrow \infty$ при $X_2 \rightarrow 0$ и/или $X_1 \rightarrow 0$; $Y_1 \rightarrow \text{const}$ с ростом X_1 и X_2) можно построить более точную формулу для аппроксимации результатов вычислительного эксперимента:

Таблица 2. Коэффициенты c_i (13)

c_0	c_1	c_2	c_3	c_4
293.3	3.083	9.0	-3.314	-2.347

$$g(X_1, X_2) = c_0 + \frac{c_1}{X_2} + \frac{c_2}{X_1} + \frac{c_3}{X_1 + X_2} + \frac{c_4}{(X_1 + X_2)^2} \quad (13)$$

где α задается в $[\text{кВт}/(\text{м}^2 \cdot \text{К})]$, u_m — в $[\text{см}/\text{с}]$, а коэффициенты c_i (табл. 2) определяются решением системы линейных уравнений

$$g(X_1, X_2) = y_i$$

для пяти пар параметров $(X_1; X_2)$: (0.5; 1), (0.5; 3), (0.5; 5), (1.5; 3), (2.5; 3) с соответствующими им значениям результатов вычислительных экспериментов y_i . На рис. 3 аппроксимация (13) показана пунктирными линиями (в виде $g - T_0$, где $T_0 = 293.3^\circ\text{C}$ — реперная температура). Погрешность этой формулы во всем параметрическом интервале не превышает 1 %, что значительно меньше погрешностей полученной ранее аппроксимации методом ОЦКП.

Отметим, что при построении интерполяции выбор набора точек, через которые будет проходить искомая функция, имеет большое значение. Если интерполяция имеет большую погрешность или СЛАУ не имеет решения, то нужно выбрать другие промежуточные параметры. Например, если зависимость нелинейная, то точка выбирается так, чтобы соответствовать не середине интервала X_i , а середине интервала Y . Для выбора набора функций, составляющего интерполяцию g , проверяются следующие условия: является ли зависимость линейной или нет; имеется ли насыщение (например, как показательная или степенная функции с отрицательным аргументом); наличие асимптот (вертикальных, горизонтальных или наклонных).

Для реализации данного метода воспользуемся программой символьных вычислений *Maxima* [6] с графическим интерфейсом *wxMaxima* [10].

Определяем функцию (13) в символьном виде

```
f1(x, y) := c0 + c1/x + c2/y + c3/(x+y) + c4/(x+y)^2;
```

и задаем наборы параметров

```
x0 : 0.5; q0 : 301.369;
x1 : 1.5; q1 : 297.544;
x2 : 2.5; q2 : 296.894;
y0 : 1.0; r0 : 302.695;
y1 : 3.0;
y2 : 5.0; r2 : 296.631;
```

Для решения СЛАУ используем команду *solve*, а для вывода результата в формате с плавающей точкой — *float*:

```

solve([f1(x0,y1)=q0,f1(x1,y0)=r0,f1(x1,y2)=r2,
f1(x1,y1)=q1,f1(x2,y1)=q2],[c0,c1,c2,c3,c4]);
float(%);

```

Более подробная информация по использованию пакета *Maxima* для решения СЛАУ, по упрощению математических выражений и построению графиков представлена в работе [7].

5. Заключение

В работе представлены примеры использования свободного ПО для обработки и анализа данных на примере двух классических методов: аппроксимация МНК и обработка результатов многопараметрических экспериментов с помощью ОЦКП. Первый метод был реализован на языке программирования *Scilab*. Было показано, что для обеспечения корректности аппроксимации исходные данные должны быть нормированы. Второй метод реализован на языке программирования C++. Приведены примеры визуализации результатов исследований с помощью программ \LaTeX и *Gnuplot*.

A Построение векторных графиков в PGFPlots

PGFPlots основан на пакетах *PGF/TikZ* и предназначен для построения двумерных и трехмерных графиков в линейном или логарифмическом масштабах с использованием различных стилей символов, линий и поверхностей. Поддерживает линейные и точечные графики, поверхности, изолинии, гистограммы, полярные координаты и многое другое.

Преимущества пакета следующие: единообразие используемого в документе и графиках типа и размера шрифта, а также стиля документа; прямое использование математического режима для описания осей; создание независимых графиков в формате *pdf* или *eps* с помощью сторонних инструментов *TikZ*.

Рассмотрим пример

```

\begin{tikzpicture}
\begin{axis}[height=6cm,width=8cm,grid=major,
font=\small,xmin=-5,xmax=5,ymin=-3,ymax=8,
legend pos=north west,
xlabel={\textcolor{red}{x}},
ylabel={\textcolor{blue}{f(x)=\left(x^2+x+
5\right)\sin(x)}}]
\addplot[orange!80!black,smooth,samples=30,
mark=square*]{4*exp(-x^2/20)*cos(deg(x))};
\addplot[blue,mark=ball]{sin(deg(x)+90)};
\legend{{f(x)=4\exp(-x^2/20)\cos(x)}, {\cos(x)}};
\end{axis}
\end{tikzpicture}

```

результат которого показан на рис. 5.

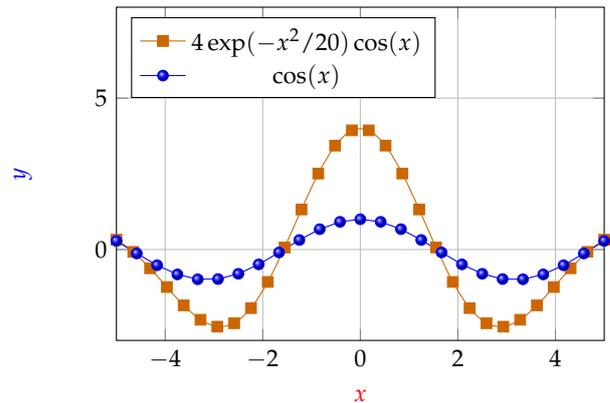


Рис. 5. Пример графика построенного в *PGFPlots*

График определяется в окружении *axis* окружения *tikzpicture*. Можно задать следующие параметры: высота *height* и ширина *width* графика; вспомогательная сетка *grid*; размер шрифта *font*; диапазон изменения значений по осям *xmin*, *xmax*, *ymin*, *ymax*; положение легенды *legend pos*; метки осей в формате \LaTeX *xlabel*, *ylabel*. Графики строятся командой *addplot*, а соответствующая графику подпись *legend*.

B Построение векторных графиков в Gnuplot

При подготовке графиков в издательской системе \LaTeX оптимальным выбором является использование *Gnuplot*-терминала *epslatex*. Данный терминал представляет собой комбинацию двух терминалов: за графическую часть отвечает *postscript eps*, а за текст и математические формулы — *latex*. Приведем пример использования данного терминала в двух режимах работы: подготовка графиков для подключения их в документ \LaTeX и подготовка независимого графического файла.

Настройка параметров графика осуществляется следующими опциями: размеры изображения — *size*, использование цветных линий и текста — *color* и *colortext*; символ десятичного разделителя — *set decimalsign*; интервал между строками в легенде — *set key spacing* (указывается, если подписи к линиям имеют большую высоту); количество точек на графике — *set samples* (для гладкого отображения быстро меняющейся функции этот параметр нужно увеличить); подписи к осям *set xlabel* и *set ylabel* могут содержать текст в формате \LaTeX (для правильной обработки такого текста он должен быть заключен в одинарные кавычки); имя выходного файла в формате \LaTeX — *set output*; формат отображения чисел на осях — *set format*; интервалы по осям *x* и *y* — *set xrange* и *set yrange*; подписи к осям — *set xtics* и

`set ytics`. Команда построения графиков `plot` может содержать несколько функций со своими параметрами отображения (тип линии — `linetype` или `lt`, тип штриховки в версии `Gnuplot` 5 и выше — `dashtype` или `dt`, толщина линии — `linewidth` или `lw`, цвет линии — `linecolor` или `lc` в формате RGB в виде названия цвета `'colorname'` или его шестнадцатеричного кода `'#RRGGBB'`; можно указать стиль линии `linestyle` или `ls`, определенный командой `set style line`), разделенными запятой. Подписи линий `title` отображаются в легенде и могут содержать текст в формате \LaTeX . Цвет фрагмента текста задается командой `\textcolor`. Для переноса длинной строки используется символ обратной косой черты `\`.

```
set terminal epslatex size 10cm,6cm\
  color colortext
set decimalsign ','
set border lw 2
set style line 1 lc rgb 'red' lt 1 dt 1 lw 5
set style line 2 lc rgb 'blue' lt 1 dt 3 lw 5
set key spacing 1.5
set samples 250
set xzeroaxis
set output 'epslatex.tex'
set xlabel 'Аргумент $x$'
set ylabel 'Функции $\textcolor{red}{f_1(x)}$ и $\textcolor{blue}{f_2(x)}$'
set format '$%g$'
set xtics ('$-3\pi$' -3*pi, '$-2\pi$' -2*pi, \
'$-\pi$' -pi, 0, '$\pi$' pi, '$2\pi$' 2*pi, \
'$3\pi$' 3*pi)
set yrange[-1:2]
plot sin(x) title '$\textcolor{red}{f_1(x)}=\sin(x)$' w l ls 1, \
sin(x**2/(pi+abs(x))+pi/2) title \
'$\textcolor{blue}{f_2(x)}=\sin\left(\frac{x^2}{\pi+|x|}+\frac{\pi}{2}\right)$' w l ls 2
```

Записываем `gnuplot`-скрипт в файл с именем `epslatex.gp` и запускаем в консоли

```
gnuplot epslatex.gp
```

При компиляции создается файл в формате \LaTeX с расширением `tex` и файл в формате `Postscript` с расширением `eps`. В документе \LaTeX изображение подключается внутри окружения `figure` командой `input` с указанием `tex`-файла. Для корректного отображения кириллицы кодировка `gnuplot`-скрипта и документа \LaTeX должны совпадать. Если предполагается создание отдельного изображения в формате `eps`, то нумерация страниц должна быть отключена, а пакеты, отвечающие за работу с текстом и формулами (например, `paratype`, `amssymb`, `amsfonts`), цветом `color` и внешними изображениями `graphicx`, подключены. Если планируется построение изолиний с метками командой `splot`, то необходимо подключить пакет `calc`. Зададим имя файла `doepslatex.tex`.

```
\documentclass{article}
\usepackage[T2A]{fontenc}
\usepackage[utf8x]{inputenc}
\usepackage[russian,english]{babel}
\usepackage{paratype}
\usepackage{amssymb}
\usepackage{amsfonts}
\usepackage{color}
\usepackage{graphicx}
\usepackage{calc}
\begin{document}
\thispagestyle{empty}
\begin{figure}
\input{epslatex}
\end{figure}
\end{document}
```

Для подготовки независимого графического файла создадим файл `compile` (в ОС Windows `compile.bat`) с командами для компиляции `gnuplot`-скрипта, компиляции файла `doepslatex.tex`, конвертации файла из формата `dvi` в формат `ps` и конвертации файла из формата `ps` в формат `eps`. В ОС GNU/Linux файл `compile` нужно сделать исполняемым (команда `chmod +x compile`).

```
gnuplot epslatex.gp
latex doepslatex.tex
dvips doepslatex.dvi
ps2eps -f doepslatex.ps
```

Результатом выполнения данного командного файла является график в файле `doepslatex.eps`, показанный на рис. 6.

Если график нужно вставить в документ, не поддерживающий формат `eps` (например, `MS Word`), то его можно преобразовать в растровый формат с помощью `GhostScript` [33] командой

```
gs -dSAFER -dBATCH -dNOPAUSE -dEPCrop -r600
-sDEVICE=png256 -sOutputFile=f.png figname.eps
```

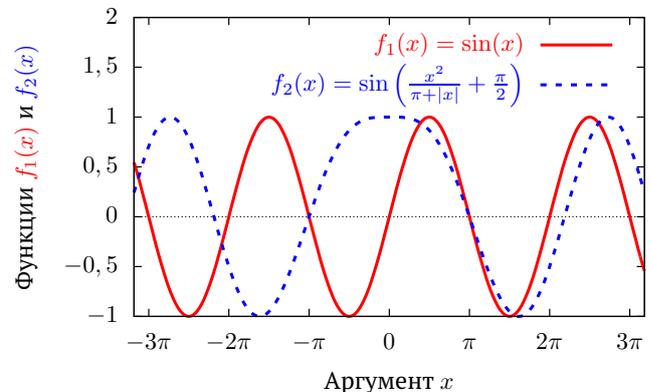


Рис. 6. Графики функций, построенные в `Gnuplot` с помощью терминала `epslatex`

Здесь используются следующие опции: *dEPSCrop* — обрезка полей в *eps*-файле; *r600* — разрешение 600 точек на дюйм; *sDEVICE* — графический формат создаваемого файла (допустимые форматы приведены на веб-сайте [33]) и его название *sOutputFile*.

Список литературы

- [1] MATLAB, MathWorks, MATLAB & Simulink. <https://www.mathworks.com/products/matlab.html> (accessed: 07.12.2021)
- [2] Scilab. <https://www.scilab.org/> (accessed: 07.12.2021)
- [3] Campbell S.L., Chancelier J.-Ph., Nikoukhah R. Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4. 2010. Springer, New York, NY. 330 p. DOI: 10.1007/978-1-4419-5527-2
- [4] GNU Octave. <http://www.gnu.org/software/octave/> (accessed: 07.12.2021)
- [5] Алексеев Е.Р., Чеснокова О.В. Введение в Octave для инженеров и математиков. М.: ALT Linux, 2012. 368 с. <http://www.altlinux.org/images/0/07/OctaveBook.pdf> (accessed: 07.12.2021)
- [6] Maxima, a computer algebra system. <https://maxima.sourceforge.io/> (accessed: 07.12.2021)
- [7] Стахин Н.А. Основы работы с системой аналитических (символьных) вычислений Maxima. Учебное пособие. М.: 2008. 86 с. http://www.uneex.ru/static/MethodBooks_Maxima/Maxima.pdf (accessed: 07.12.2021)
- [8] Wolfram Mathematica: Modern Technical Computing. <http://www.wolfram.com/mathematica> (accessed: 07.12.2021)
- [9] Maplesoft: software for mathematics, online learning, engineering. <http://www.maplesoft.com/> (accessed: 07.12.2021)
- [10] wxMaxima. <https://wxmaxima-developers.github.io/wxmaxima/> (accessed: 07.12.2021)
- [11] Насибуллаев И.Ш. Применение свободных программ FreeFem++/Gmsh и FreeCAD/CalculX для моделирования статических структурных задач // Многофазные системы. 2020. Т. 15, № 3–4. С. 183–200. DOI: 10.21662/mfs2020.3.129
- [12] Bretscher O. Linear Algebra With Applications (3rd ed.). 1995. Upper Saddle River, NJ: Prentice Hall.
- [13] Durakovic B. Design of Experiments Application, Concepts, Examples: State of the Art. // Periodicals of Engineering and Natural Sciences. 2017. Vol. 5, No. 3. Pp. 421–439. DOI: 10.21533/pen.v5i3.145
- [14] Адлер Ю.П., Маркова Е.В., Грановский Ю.В. Планирование эксперимента при поиске оптимальных условий. М.: Наука, 1976. 280 с.
- [15] LaTeX — a document preparation system. <http://www.latex-project.org/> (accessed: 07.12.2021)
- [16] Котельников И.А., Чеботаев П.З. LaTeX по-русски. СПб.: «Корона-Век». 2011. 496 с.
- [17] van Dongen M.R.C. LaTeX and Friends. 2012. Springer-Verlag. Berlin, Heidelberg. 300 p. DOI: 10.1007/978-3-642-23816-1
- [18] Microsoft. <https://www.microsoft.com/> (accessed: 07.12.2021)
- [19] Feuersänger C. Manual for Package PGFPlots, 2D/3D Plots in LATEX, Version 1.18. <http://pgfplots.sourceforge.net/pgfplots.pdf> (accessed: 07.12.2021)
- [20] The TikZ and PGF Packages. Manual for version 3.1.9a. Ed. by Till Tantau. Universität zu Lübeck, 2021. <https://mirror.truenetwork.ru/CTAN/graphics/pgf/base/doc/pgfmanual.pdf> (accessed: 07.12.2021)
- [21] PSTricks. <https://tug.org/PSTricks/main.cgi> (accessed: 07.12.2021)
- [22] Gnuplot. <http://www.gnuplot.info/> (accessed: 07.12.2021)
- [23] Origin — data analysis and graphing software. <http://www.originlab.com/> (accessed: 07.12.2021)
- [24] CTAN: package listings. <https://ctan.org/pkg/listings> (accessed: 07.12.2021)
- [25] CTAN: package beamer. <https://ctan.org/pkg/beamer> (accessed: 07.12.2021)
- [26] Насибуллаев И.Ш. Аналитический анализ переключения рабочего режима в двумерной модели системы жидкостного охлаждения микрозахвата // Вестник УГАТУ. 2021. Т. 25, № 3 (93). С. 120–131. DOI: 10.54708/19926502_2021_25393120
- [27] Волков А.И., Жарский И.М. Большой химический справочник. М.: Современная школа. 2005. 608 с.
- [28] Khuri A.I., Cornell J.A. Response surface, design and analysis, 2nd edition, Marcel Dekker Inc., New York. 1996.
- [29] Насибуллаев И.Ш. Разработка компьютерной модели основного элемента агрегата дозирования топлива // Вычислительные технологии. 2016. Т. 21, № 2. С. 26–41. <http://www.ict.nsc.ru/jct/getfile.php?id=1724>
- [30] Oyejola B.A., Nwanya J.C. Selecting the Right Central Composite Design. International Journal of Statistics and Applications. 2015. Vol. 5, No. 1. Pp. 21–30. DOI: 10.5923/j.statistics.20150501.04
- [31] Насибуллаев И.Ш., Даринцев О.В. Компьютерное двумерное моделирование системы жидкостного охлаждения микрозахвата // Вычислительные технологии. 2021. Т. 26, № 2. С. 4–20. DOI: 10.25743/ICT.2021.26.2.002
- [32] Шкляр В.Н. Планирование эксперимента и обработка результатов. Издательство Томского политехнического университета. 2010.
- [33] Ghostscript. <https://www.ghostscript.com/> (accessed: 07.12.2021)



Application of free software for processing and visualization of scientific research results

Nasibullayev I.Sh.

Mavlyutov Institute of Mechanics UFRC RAS, Ufa, Russia

The paper presents an overview and examples of the use of free software for processing, analyzing and visualizing the results of computational and natural experiments. The implementation of two standard methods of approximation and data analysis is considered: the method of least squares and orthogonal central composite design. The first method allows the approximation of tabular data in the form of a polynomial of arbitrary degree. With an increase in the degree of the polynomial, the accuracy of the approximation increases, but, due to the growth of the condition number of the matrix, the correctness of the approximation decreases. It is shown that the normalization of the initial data makes it possible to increase the maximum order of the polynomial, in which there will be no loss of accuracy due to an ill-conditioned matrix. The implementation of the method in the Scilab programming language is proposed. The second method makes it possible to obtain a second-order approximation for an arbitrary number of variables, to estimate the linear and quadratic contributions of individual variables and their coupled interactions. A description of the method is presented and a scheme for constructing an extended experiment design matrix is proposed. The method is implemented in the C++ programming language, and a script written in the Scilab language is proposed to determine the error of the obtained approximation. To obtain a more accurate approximation, one can choose a function that takes into account the asymptotic features of the results of the computational experiment. An example of such an approximation is implemented in the computer algebra software wxMaxima. The presented software implementations of approximation methods additionally generate code in the Gnuplot and Scilab format, which increases efficiency for subsequent visualization or analysis of approximations. Examples of research visualization in the PGFPlots \LaTeX package and in the Gnuplot program using the epslatex terminal are given. Both approaches are convenient when presenting research results in the \LaTeX desktop publishing system.

Keywords: free software, the method of least squares, orthogonal central composite design, Scilab, wxMaxima, Latex, Gnuplot

References

- [1] MATLAB, MathWorks, MATLAB & Simulink.
<https://www.mathworks.com/products/matlab.html>
(accessed: 07.12.2021)
- [2] Scilab.
<https://www.scilab.org/>
(accessed: 07.12.2021)
- [3] Campbell S.L., Chancelier J-Ph., Nikoukhah R. Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4. 2010. Springer, New York, NY. 330 p.
DOI: 10.1007/978-1-4419-5527-2
- [4] GNU Octave.
<http://www.gnu.org/software/octave/>
(accessed: 07.12.2021)
- [5] Alekseyev Ye.R., Chesnokova O.V. [Introduction to Octave for engineers and mathematicians] Vvedeniye v Octave dlya inzhenerov i matematikov. M.: ALT Linux, 2012. 368 p. (in Russian).
<http://www.altlinux.org/images/0/07/OctaveBook.pdf>
(accessed: 07.12.2021)
- [6] Maxima, a computer algebra system.
<https://maxima.sourceforge.io/>
(accessed: 07.12.2021)
- [7] Stakhin N.A. [Fundamentals of working with the system of analytical (symbolic) calculations Maxima. Tutorial] Osnovy raboty s sistemoy analiticheskikh (simvolnykh) vychisleniy Maxima. Uchebnoye posobiye. M.: 2008. 86 p. (in Russian).
http://www.uneex.ru/static/MethodBooks_Maxima/Maxima.pdf
(accessed: 07.12.2021)
- [8] Wolfram Mathematica: Modern Technical Computing.
<http://www.wolfram.com/mathematica>
(accessed: 07.12.2021)
- [9] Maplesoft: software for mathematics, online learning, engineering.
<http://www.maplesoft.com/> (accessed: 07.12.2021)

- [10] wxMaxima.
<https://wxmaxima-developers.github.io/wxmaxima/>
(accessed: 07.12.2021)
- [11] Nasibullayev I.Sh. [Application of free software FreeFem++/Gmsh and FreeCAD/CalculiX for simulation of static elasticity problems] *Primeneniye svobodnykh programm FreeFem++/Gmsh i FreeCAD/CalculiX dlya modelirovaniya sticheskiikh strukturnykh zadach // Mnogofaznyye sistemy [Multiphase Systems]*. 2020. Vol. 15, N 3–4. Pp. 183–200. (in Russian).
DOI: 10.21662/mfs2020.3.129
- [12] Bretscher O. *Linear Algebra With Applications* (3rd ed.). 1995. Upper Saddle River, NJ: Prentice Hall.
- [13] Durakovic B. *Design of Experiments Application, Concepts, Examples: State of the Art. // Periodicals of Engineering and Natural Sciences*. 2017. Vol. 5, No. 3. Pp. 421–439.
DOI: 10.21533/pen.v5i3.145
- [14] Adler Yu.P., Markova Ye.V., Granovskiy Yu.V. [Design of experiment in the search for optimal conditions] *Planirovaniye eksperimenta pri poiske optimalnykh usloviy*. M.: Nauka, 1976. 280 p. (in Russian).
- [15] LaTeX – a document preparation system.
<http://www.latex-project.org/>
(accessed: 07.12.2021)
- [16] Kotelnikov I.A., Chebotayev P.Z. [LaTeX in Russian] *LaTeX po-russki*. SPb.: “Korona-Vek”. 2011. 496 p. (in Russian).
- [17] van Dongen M.R.C. *LaTeX and Friends*. 2012. Springer-Verlag, Berlin, Heidelberg. 300 p. (in Russian).
DOI: 10.1007/978-3-642-23816-1
- [18] Microsoft.
<https://www.microsoft.com/>
(accessed: 07.12.2021)
- [19] Feuersänger C. *Manual for Package PGFPlots, 2D/3D Plots in LATEX, Version 1.18*.
<http://pgfplots.sourceforge.net/pgfplots.pdf>
(accessed: 07.12.2021)
- [20] *The TikZ and PGF Packages. Manual for version 3.1.9a*. Ed. by Till Tantau. Universität zu Lübeck, 2021.
<https://mirror.truenetwork.ru/CTAN/graphics/pgf/base/doc/pgfmanual.pdf>
(accessed: 07.12.2021)
- [21] PSTricks.
<https://tug.org/PSTricks/main.cgi>
(accessed: 07.12.2021)
- [22] Gnuplot.
<http://www.gnuplot.info/>
(accessed: 07.12.2021)
- [23] Origin – data analysis and graphing software.
<http://www.originlab.com/>
(accessed: 07.12.2021)
- [24] CTAN: package listings.
<https://ctan.org/pkg/listings>
(accessed: 07.12.2021)
- [25] CTAN: package beamer.
<https://ctan.org/pkg/beamer>
(accessed: 07.12.2021)
- [26] Nasibullayev I.Sh. [Analytical analysis of operating mode switching in a 2D model of a fluid cooling system of the micro-gripper] *Analiticheskiy analiz pereklyucheniya rabocheho rezhima v dvumernoy modeli sistemy zhidkostnogo okhlazhdeniya mikrozhkhvata // Vestnik UGATU*. 2021. Vol. 25, N. 3 (93). Pp. 120–131. (in Russian).
DOI: 10.54708/19926502_2021_25393120
- [27] Volkov A.I., Zharskiy I.M. [Big chemical reference book] *Bolshoy khimicheskiy spravochnik*. M.: Sovremennaya shkola [Modern school]. 2005. 608 p. (in Russian).
- [28] Khuri A.I., Cornell J.A. *Response surface, design and analysis*, 2nd edition, Marcel Dekker Inc., New York. 1996.
- [29] Nasibullayev I.Sh. [The development of a computer model for the main element of the fuel metering unit] *Razrabotka kompyuternoy modeli osnovnogo elementa agregata dozirovaniya topliva. Vychislitelnye tehnologii [Computational Technologies]*. 2016. V. 21, No. 2. Pp. 26–41. (In Russian).
<http://www.ict.nsc.ru/jct/getfile.php?id=1724>
- [30] Oyejola B.A., Nwanya J.C. *Selecting the Right Central Composite Design. International Journal of Statistics and Applications*. 2015. Vol. 5, No. 1. Pp. 21–30.
DOI: 10.5923/j.statistics.20150501.04
- [31] Nasibullayev I.Sh., Darintsev O.V. [Computer 2D modelling of a micro-grip fluid cooling system] *Komp'yuternoye dvumernoye modelirovaniye sistemy zhidkostnogo okhlazhdeniya mikrozhkhvata // Vychislitelnye tehnologii [Computational Technologies]*. 2021. Vol. 26, N. 2. Pp. 4–20. (In Russian).
DOI: 10.25743/ICT.2021.26.2.002
- [32] Shklyar V.N. [Design of experiment and results processing] *Planirovaniye eksperimenta i obrabotka rezultatov*. Izdatelstvo Tomskogo politekhnicheskogo universiteta [Tomsk Polytechnic University Publishing House]. 2010. (In Russian).
- [33] Ghostscript.
<https://www.ghostscript.com/>
(accessed: 07.12.2021)