НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

ПРИ НАЦІОНАЛЬНОМУ ТЕХНІЧНОМУ УНІВЕРСИТЕТІ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Лабораторна робота №3

з системного аналізу

**«Відновлення функціональних залежностей в мультиплікативній формі по дискретній вибірці»**

Варіант № 2

Виконали:

студенти групи КА-12

Аввакумов Тарас

Трусковський Кирил

Федь Володимир

Київ, 2014

### Постановка задачи

1. Построить по заданной дискретной выборке (тестовая выборка приведена в таблице) и для реальной физической задачи оценивания составляющих солнечных бурь **Dst**  в мультипликативной форме приближающие функции ,  (аналитически и графически представленные функциональные зависимости), которые с практически приемлемой погрешностью в смысле Чебышевского приближения характеризуют истинные функциональные зависимости , .

Для функций  известны границы интервалов, определяемые следующими условиями:

 ;

.

Для функций  известны границы интервалов, определяемые следующими условиями:



; 

1. Построить прогнозные значения
2. Предложить свой вариант дискретной выборки и  (для реальной задачи) и построить в мультипликативной форме приближающие функции , .
3. Предложить свой вариант структуры функций ,  и построить в мультипликативной форме приближающие функции , 
4. Провести обзор и привести имеющуюся литературу по данному вопросу,

**Таблица № 1**(В соответствии с нашим вариантом)

Данные с 23 по 25 3екабря 1995 года за каждый час, начало 12 часов дня 23 3екабря, конец - 11 часов дня 25 мая

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | X1=[Bx, By] | | X2=[V/1000, Bz] | | X3=[Dst(k-1), Dst(k-2)] | |
| N | Y1=Dst(k) | Y2=Dst(k+1) |  | X11=Bx | X12=By | X21=V/1000 | X22=Bz | X31=Dst(k-1) | X32=Dst(k-2) |
| 1 | -19 | -18 |  | -1.2 | -3.9 | 0.3590 | -1.2 | -26 | -30 |
| 2 | -18 | -20 |  | -1.4 | -3.8 | 0.3920 | -1.6 | -19 | -26 |
| 3 | -20 | -19 |  | -1.6 | -2.7 | 0.4020 | 0.8 | -18 | -19 |
| 4 | -19 | -19 |  | -3.0 | -1.2 | 0.3940 | 0.8 | -20 | -18 |
| 5 | -19 | -19 |  | -2.9 | -0.9 | 0.4000 | -0.2 | -19 | -20 |
| 6 | -19 | -18 |  | -4.1 | -0.5 | 0.3920 | -0.5 | -19 | -19 |
| 7 | -18 | -15 |  | 0.1 | -2.2 | 0.3980 | -0.2 | -19 | -19 |
| 8 | -15 | -12 |  | 1.5 | -1.7 | 0.4090 | 0.6 | -18 | -19 |
| 9 | -12 | -11 |  | 3.4 | -3.0 | 0.3850 | 0.4 | -15 | -18 |
| 10 | -11 | -13 |  | -0.2 | -1.5 | 0.3800 | 1.9 | -12 | -15 |
| 11 | -13 | -11 |  | 3.3 | -4.0 | 0.3690 | 2.7 | -11 | -12 |
| 12 | -11 | -7 |  | 1.0 | -6.2 | 0.3660 | 0.3 | -13 | -11 |
| 13 | -7 | -5 |  | 4.0 | -4.9 | 0.3620 | 0.2 | -11 | -13 |
| 14 | -5 | -4 |  | 2.4 | -3.0 | 0.3550 | 0.2 | -7 | -11 |
| 15 | -4 | -3 |  | -3.7 | 4.1 | 0.3460 | 0.7 | -5 | -7 |
| 16 | -3 | -4 |  | -2.6 | 3.2 | 0.3440 | -4.8 | -4 | -5 |
| 17 | -4 | -3 |  | -3.7 | 5.9 | 0.3400 | 0.1 | -3 | -4 |
| 18 | -3 | 28 |  | -3.4 | 4.7 | 0.3340 | 0.0 | -4 | -3 |
| 19 | 28 | 16 |  | -4.9 | 6.6 | 0.4020 | -1.4 | -3 | -4 |
| 20 | 16 | -6 |  | -3.8 | 10.9 | 0.4330 | -9.9 | 28 | -3 |
| 21 | -6 | -18 |  | -6.5 | 3.7 | 0.4430 | 4.1 | 16 | 28 |
| 22 | -18 | -31 |  | 2.1 | -1.5 | 0.4910 | -8.9 | -6 | 16 |
| 23 | -31 | -29 |  | -9.6 | 8.4 | 0.5260 | 7.5 | -18 | -6 |
| 24 | -29 | -45 |  | -3.8 | 4.1 | 0.5200 | -3.1 | -31 | -18 |
| 25 | -45 | -55 |  | -1.0 | 4.7 | 0.5350 | -4.1 | -29 | -31 |
| 26 | -55 | -61 |  | 2.2 | 6.7 | 0.5480 | -4.0 | -45 | -29 |
| 27 | -61 | -65 |  | -2.2 | 4.4 | 0.5620 | -2.3 | -55 | -45 |
| 28 | -65 | -60 |  | -1.7 | 1.6 | 0.5740 | 0.9 | -61 | -55 |
| 29 | -60 | -56 |  | 3.5 | -0.9 | 0.5600 | 1.0 | -65 | -61 |
| 30 | -56 | -50 |  | 3.8 | 0.7 | 0.5390 | 2.1 | -60 | -65 |
| 31 | -50 | -52 |  | 1.7 | 2.0 | 0.5320 | -0.5 | -56 | -60 |
| 32 | -52 | -48 |  | -0.5 | -0.5 | 0.5490 | 0.1 | -50 | -56 |
| 33 | -48 | -46 |  | 1.3 | -3.9 | 0.5640 | -1.9 | -52 | -50 |
| 34 | -46 | -42 |  | -1.7 | 2.0 | 0.5980 | 1.3 | -48 | -52 |
| 35 | -42 | -42 |  | 2.8 | 2.9 | 0.6070 | 0.1 | -46 | -48 |
| 36 | -42 | -39 |  | 0.1 | 3.6 | 0.5890 | -0.4 | -42 | -46 |
| 37 | -39 | -38 |  | -0.5 | 4.6 | 0.6140 | 1.4 | -42 | -42 |
| 38 | -38 | -32 |  | -2.2 | 4.5 | 0.5940 | 1.5 | -39 | -42 |
| 39 | -32 | -31 |  | 0.0 | 2.4 | 0.6210 | 0.4 | -38 | -39 |
| 40 | -31 | -30 |  | -0.9 | 3.6 | 0.5940 | -0.3 | -32 | -38 |
| 41 | -30 | -25 |  | 1.8 | 1.9 | 0.5990 | -0.4 | -31 | -32 |
| 42 | -25 | -22 |  | -1.6 | 3.2 | 0.5890 | -0.4 | -30 | -31 |
| 43 | -22 | -22 |  | -3.4 | 2.3 | 0.6060 | 0.0 | -25 | -30 |
| 44 | -22 | -20 |  | -3.8 | -0.7 | 0.6540 | 2.6 | -22 | -25 |
| 45 | -20 | -22 |  | -4.2 | -1.4 | 0.6580 | 4.3 | -22 | -22 |
| 46 | -22 | -28 |  | -4.2 | 0.6 | 0.6730 | 1.1 | -20 | -22 |
| 47 | -28 | -28 |  | -5.6 | 0.3 | 0.6570 | 1.7 | -22 | -20 |
| 48 | -28 | -25 |  | -4.5 | 1.6 | 0.6520 | 0.2 | -28 | -22 |

**Таблица №2** тестовых исходных данных для всех вариантов

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 5,05 | 8,65 | 7,75 | 6,975 | 4,879 | 3,501 | 5.967 | 254,621 | 98,145 | 119,406 | 117,683 |
| 2 | 5,052 | 8,7 | 7,78 | 6.955 | 4,886 | 3,553 | 5,978 | 198,163 | 73,368 | 92,651 | 90,123 |
| 3 | 5,055 | 8,745 | 7,80 | 6,95 | 4,897 | 3,611 | 5,984 | 187,411 | 91,084 | 87,691 | 83,576 |
| 4 | 5,06 | 8,75 | 7,82 | 6,945 | 4,916 | 3,652 | 5,987 | 167,197 | 123,567 | 78,793 | 74,789 |
| 5 | 5,063 | 9,8 | 7,845 | 6,925 | 4,938 | 3,723 | 5, 996 | 166,547 | 163,813 | 79,497 | 74,316 |
| 6 | 5064 | 10,25 | 7,851 | 6,895 | 4,947 | 3,758 | 5, 999 | 153,789 | 261,378 | 77,082 | 72,817 |
| 7 | 5,067 | 11,85 | 7,852 | 6,865 | 4,954 | 3,784 | 5,976 | 110,926 | 355,579 | 67,758 | 77,425 |
| 8 | 5,07 | 12,87 | 7,853 | 6,854 | 4.967 | 3,809 | 5,964 | 151,381 | 440,432 | 51,956 | 89,519 |
| 9 | 5,075 | 14,90 | 8,854 | 6,856 | 4,978 | 3,825 | 5,958 | 187,364 | 336,283 | 91,123 | 121,374 |
| 10 | 5,08 | 16,91 | 8,855 | 6,855 | 4,984 | 3,845 | 5,937 | 236,123 | 223,657 | 112,859 | 149,173 |
| 11 | 5,085 | 18,92 | 9,856 | 6,856 | 4,987 | 3,851 | 5,916 | 292,341 | 118,624 | 153,717 | 184,136 |
| 12 | 5,09 | 15,92 | 10,86 | 6,865 | 4, 996 | 3,8534 | 5,874 | 344,324 | 91,324 | 117,965 | 179,152 |
| 13 | 5,095 | 12,93 | 11,85 | 7,859 | 4 999 | 3,8536 | 5,842 | 426,939 | 68,926 | 155,912 | 201,239 |
| 14 | 5,1 | 11,93 | 12,87 | 7,876 | 4,976 | 3,854 | 5,814 | 477,128 | 44,675 | 169,359 | 225,482 |
| 15 | 5,125 | 9,935 | 11,89 | 7,895 | 4.964 | 3,856 | 5,756 | 505,327 | 29,367 | 192,924 | 240,976 |
| 16 | 5,135 | 8,941 | 9,925 | 7925 | 4,958 | 3,859 | 5,718 | 558,386 | 18,567 | 218,549 | 275,846 |
| 17 | 5,15 | 7,945 | 8,945 | 7,945 | 4,937 | 3,867 | 5,671 | 618,859 | 23,932 | 247,354 | 316,124 |
| 18 | 5,153 | 6,951 | 7,945 | 7,951 | 4,916 | 3,879 | 5,629 | 895,737 | 35,124 | 284,167 | 363,928 |
| 19 | 5,157 | 5,965 | 6,95 | 6.955 | 4,874 | 3,886 | 5,567 | 906,168 | 61,946 | 316,375 | 403,153 |
| 20 | 5,2 | 4,965 | 5.965 | 6,975 | 4,842 | 3,897 | 5,486 | 885,761 | 121,387 | 341,326 | 431,195 |
| 21 | 5,25 | 3,974 | 4.975 | 7,001 | 4,814 | 3,916 | 5,452 | 790,639 | 310,519 | 375,651 | 471,588 |
| 22 | 5,3 | 2,981 | 5,000 | 7,125 | 4,756 | 3,938 | 5,501 | 723,784 | 485,142 | 446,856 | 436,847 |
| 23 | 5,315 | 3,985 | 6,975 | 7,145 | 4,718 | 3,947 | 5,554 | 731,438 | 588,125 | 548,314 | 441,842 |
| 24 | 5,325 | 4990 | 7.955 | 7,165 | 4,671 | 3,954 | 5,621 | 721,321 | 683,435 | 644,716 | 439,425 |
| 25 | 5,35 | 5,995 | 8,945 | 7,195 | 4,629 | 3,.967 | 5,658 | 691,845 | 772,834 | 729,942 | 422,147 |
| 26 | 5,353 | 7,997 | 9,935 | 7,209 | 4,567 | 3,978 | 5,712 | 508,614 | 880,562 | 849,316 | 435,954 |
| 27 | 5,357 | 9,001 | 10,92 | 7,225 | 4,482 | 3,984 | 5,753 | 429,956 | 687,987 | 748,231 | 450,492 |
| 28 | 5,4 | 10,94 | 11,89 | 7,25 | 4,452 | 3,987 | 5,781 | 330,129 | 488,951 | 647,987 | 454,897 |
| 29 | 5,425 | 12,90 | 12,86 | 7,975 | 4,364 | 3, 996 | 5,802 | 127,152 | 385,494 | 442,967 | 458,289 |
| 30 | 5,445 | 10,88 | 14,85 | 7,955 | 4,326 | 3, 999 | 5,825 | 78,654 | 211,209 | 232,856 | 172,164 |
| 31 | 5,465 | 8,944 | 15,85 | 7,95 | 4,264 | 3,976 | 5,845 | 52,145 | 196,197 | 115,632 | 153,356 |
| 32 | 5,475 | 6,780 | 12,85 | 7,945 | 4,184 | 3.964 | 5,851 | 86,243 | 87,325 | 93,135 | 127,168 |
| 33 | 5485 | 6,764 | 10,85 | 7,925 | 4,156 | 3,958 | 5,854 | 126,345 | 64,615 | 77,824 | 106,123 |
| 34 | 5,495 | 6,568 | 8,865 | 7,895 | 4,136 | 3,937 | 5,856 | 132,879 | 52,534 | 63,453 | 82,659 |
| 35 | 5,497 | 6,437 | 6,859 | 7,865 | 4,129 | 3,916 | 5,854 | 167,156 | 32,178 | 52,167 | 93,834 |
| 36 | 5,5 | 5325 | 4,876 | 7,854 | 4,116 | 3,874 | 5,856 | 170,531 | 66,176 | 42,836 | 91,345 |
| 37 | 5515 | 5,206 | 2,895 | 7,853 | 4,098 | 3,842 | 5,859 | 184,243 | 70,364 | 37,192 | 96,841 |
| 38 | 5,525 | 5,149 | 1,925 | 7,855 | 4,0816 | 3,814 | 5,867 | 191,956 | 76,428 | 25,834 | 93,952 |
| 39 | 5,545 | 5.089 | 3,945 | 7,856 | 4,0686 | 3,756 | 5,879 | 216,829 | 83,475 | 50,985 | 109,463 |
| 40 | 5,575 | 4,933 | 4,953 | 7,865 | 4,0486 | 3,718 | 5,886 | 383,329 | 104,924 | 98,591 | 133,415 |
| 41 | 5,6 | 4,889 | 5.955 | 7,859 | 4,0246 | 3,671 | 5,005 | 279,421 | 184,183 | 102,861 | 108,613 |
| 42 | 5,65 | 3,935 | 6,975 | 7,876 | 4,0126 | 3,629 | 5,027 | 225,356 | 286,324 | 105,817 | 107,319 |
| 43 | 5,7 | 3,941 | 7.001 | 7,895 | 4,0114 | 3,567 | 5,049 | 176,578 | 366,457 | 78,473 | 82,263 |
| 44 | 5,745 | 2,945 | 7,125 | 7,925 | 4,0026 | 3,484 | 5,095 | 170,948 | 265,814 | 81,417 | 84,132 |
| 45 | 5,75 | 3,95 | 7,145 | 7,945 | 4,0019 | 3,452 | 5,189 | 158,334 | 184,549 | 78,653 | 81,953 |

# Варианты структуры функций

Функциональные зависимости формируются в классе мультипликативных функций, которые характеризуются последовательностью следующих уровней

 ; (1)

; (2)

; (3)

. (4)

Данная последовательность для удобства вычислений после несложных преобразований представляется в следующей форме аддитивных функций

 ;  ; ; (5)

**; ;** (6)

**;**  . (7)

**Варианты функций** 

 ****; 

 ****; 

 ****; 

**1.**

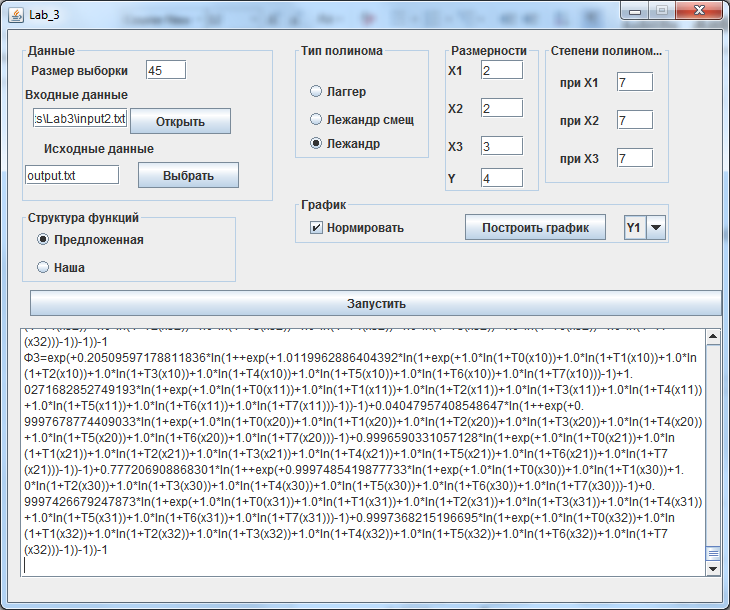
1.1   
Построим по заданной **дискретной выборке** в мультипликативной форме приближающие функции , 

***1 вид:***   
Полиномы Лежандра :



 но 

 ; 



psi010=exp(+1.0078765932605016\*ln(1+T0(x10))+-9.375095617830254\*ln(1+T1(x10))+4.437805621241115\*ln(1+T2(x10))+-1.2691495973994007\*ln(1+T3(x10))+1.1966761918721602\*ln(1+T4(x10))+-1.5341251988476605\*ln(1+T5(x10))+1.237255058372709\*ln(1+T6(x10))+-0.4368878758516289\*ln(1+T7(x10)))-1

psi011=exp(+1.0078765932605016\*ln(1+T0(x11))+5.17384452598994\*ln(1+T1(x11))+-1.819836935542979\*ln(1+T2(x11))+-1.4549071149564028\*ln(1+T3(x11))+2.9345179308586515\*ln(1+T4(x11))+-1.6642417955025495\*ln(1+T5(x11))+-0.940641848314842\*ln(1+T6(x11))+1.197747428461699\*ln(1+T7(x11)))-1

psi020=exp(+-0.25380979664043984\*ln(1+T0(x20))+0.4289045094610338\*ln(1+T1(x20))+0.7227283502179692\*ln(1+T2(x20))+-1.2989615969240558\*ln(1+T3(x20))+0.9008468089018122\*ln(1+T4(x20))+-0.3117120624809128\*ln(1+T5(x20))+-0.7341894036978045\*ln(1+T6(x20))+0.707519643773089\*ln(1+T7(x20)))-1

psi021=exp(+-0.25380979664043984\*ln(1+T0(x21))+-0.04675973354969946\*ln(1+T1(x21))+0.5226773495857509\*ln(1+T2(x21))+-1.7188528534431664\*ln(1+T3(x21))+2.0248058364011423\*ln(1+T4(x21))+-1.0495094035322556\*ln(1+T5(x21))+-0.16785532583648377\*ln(1+T6(x21))+0.20500566867309422\*ln(1+T7(x21)))-1

psi030=exp(+86423.03413424565\*ln(1+T0(x30))+-568.4563489162756\*ln(1+T1(x30))+304.01062413800054\*ln(1+T2(x30))+-68.39947832130575\*ln(1+T3(x30))+6.133133195584433\*ln(1+T4(x30))+0.6469420412788801\*ln(1+T5(x30))+-0.4908310171850466\*ln(1+T6(x30))+0.2660481857051654\*ln(1+T7(x30)))-1

psi031=exp(+86423.03413424565\*ln(1+T0(x31))+-86341.27009906256\*ln(1+T1(x31))+266439.6092565686\*ln(1+T2(x31))+-73379.06636020869\*ln(1+T3(x31))+-115616.57812721531\*ln(1+T4(x31))+-80417.9218773619\*ln(1+T5(x31))+-112044.95377087317\*ln(1+T6(x31))+-57655.34547189864\*ln(1+T7(x31)))-1

psi032=exp(+86423.03413424565\*ln(1+T0(x32))+154.2937517115147\*ln(1+T1(x32))+-94.3141571903461\*ln(1+T2(x32))+92.81059234963178\*ln(1+T3(x32))+-73.60407328539421\*ln(1+T4(x32))+-10.27566749418246\*ln(1+T5(x32))+24.46216126980048\*ln(1+T6(x32))+-1.977703102877775\*ln(1+T7(x32)))-1

psi110=exp(+3.3344016821136573\*ln(1+T0(x10))+9.461107273638747\*ln(1+T1(x10))+-5.176099921515887\*ln(1+T2(x10))+2.5498220661342472\*ln(1+T3(x10))+-2.0354403894204265\*ln(1+T4(x10))+1.260951903109428\*ln(1+T5(x10))+-0.664840028133066\*ln(1+T6(x10))+0.06851358705174758\*ln(1+T7(x10)))-1

psi111=exp(+3.3344016821136573\*ln(1+T0(x11))+-17.816737303546205\*ln(1+T1(x11))+7.517411166837675\*ln(1+T2(x11))+0.6224149969389731\*ln(1+T3(x11))+-0.9352015553388657\*ln(1+T4(x11))+-3.0918415974612015\*ln(1+T5(x11))+6.2628474783546775\*ln(1+T6(x11))+-3.72304083703611\*ln(1+T7(x11)))-1

psi120=exp(+0.12074874102958975\*ln(1+T0(x20))+0.012912051583008191\*ln(1+T1(x20))+-0.10779750487948979\*ln(1+T2(x20))+0.06551890720373688\*ln(1+T3(x20))+0.028817604548443364\*ln(1+T4(x20))+0.04504370027888293\*ln(1+T5(x20))+-0.033977997794590216\*ln(1+T6(x20))+-0.12144092008463372\*ln(1+T7(x20)))-1

psi121=exp(+0.12074874102958975\*ln(1+T0(x21))+0.07837474533891126\*ln(1+T1(x21))+-0.03462132190849608\*ln(1+T2(x21))+-0.11493689998702729\*ln(1+T3(x21))+-0.17955323271276938\*ln(1+T4(x21))+-0.02396041907274321\*ln(1+T5(x21))+0.06345518029529648\*ln(1+T6(x21))+0.10100055464722714\*ln(1+T7(x21)))-1

psi130=exp(+-1.5975054115402938\*ln(1+T0(x30))+4.762336376385421\*ln(1+T1(x30))+8.386999722312005\*ln(1+T2(x30))+-11.00044597032626\*ln(1+T3(x30))+0.20550284631159127\*ln(1+T4(x30))+10.584559205144119\*ln(1+T5(x30))+-13.70780546917582\*ln(1+T6(x30))+7.047799857475385\*ln(1+T7(x30)))-1

psi131=exp(+-1.5975054115402938\*ln(1+T0(x31))+6.464971342014239\*ln(1+T1(x31))+2.980242231078077\*ln(1+T2(x31))+-8.035874269878185\*ln(1+T3(x31))+-0.4022768792502549\*ln(1+T4(x31))+11.522586572836381\*ln(1+T5(x31))+2.0146703635742957\*ln(1+T6(x31))+-12.03635048231135\*ln(1+T7(x31)))-1

psi132=exp(+-1.5975054115402938\*ln(1+T0(x32))+-3.9039023114795564\*ln(1+T1(x32))+-2.100995583127316\*ln(1+T2(x32))+7.727994482336271\*ln(1+T3(x32))+7.128648036353176\*ln(1+T4(x32))+1.6304309695128028\*ln(1+T5(x32))+4.186026034933914\*ln(1+T6(x32))+-7.516694389282384\*ln(1+T7(x32)))-1

psi210=exp(+-0.22056934268672942\*ln(1+T0(x10))+-1.3078134486919297\*ln(1+T1(x10))+0.34944768454305347\*ln(1+T2(x10))+1.2627586180426957\*ln(1+T3(x10))+-1.5936442015625891\*ln(1+T4(x10))+0.9505741542115419\*ln(1+T5(x10))+0.6797153789697775\*ln(1+T6(x10))+-0.9827966299625795\*ln(1+T7(x10)))-1

psi211=exp(+-0.22056934268672942\*ln(1+T0(x11))+3.4558082132958186\*ln(1+T1(x11))+-1.9181805223484345\*ln(1+T2(x11))+0.9682600258074707\*ln(1+T3(x11))+-0.5263606700339348\*ln(1+T4(x11))+-0.020690208490231236\*ln(1+T5(x11))+0.44779865216838816\*ln(1+T6(x11))+-0.309610313560924\*ln(1+T7(x11)))-1

psi220=exp(+-0.37845985682442645\*ln(1+T0(x20))+2.9097907299936243\*ln(1+T1(x20))+-1.504532983163441\*ln(1+T2(x20))+0.41364836780824965\*ln(1+T3(x20))+-0.2116645079264474\*ln(1+T4(x20))+0.7765563448716019\*ln(1+T5(x20))+-1.113143121489567\*ln(1+T6(x20))+0.7292271869541748\*ln(1+T7(x20)))-1

psi221=exp(+-0.37845985682442645\*ln(1+T0(x21))+-1.0748385375915812\*ln(1+T1(x21))+0.27361361685452656\*ln(1+T2(x21))+0.9340071396232857\*ln(1+T3(x21))+-1.8212287016937194\*ln(1+T4(x21))+1.442358704499684\*ln(1+T5(x21))+-0.3030399421439247\*ln(1+T6(x21))+-0.06306147162419234\*ln(1+T7(x21)))-1

psi230=exp(+-25833.867530853386\*ln(1+T0(x30))+-1276.7928585315635\*ln(1+T1(x30))+648.4858059614834\*ln(1+T2(x30))+-220.09014215052113\*ln(1+T3(x30))+119.89947402400779\*ln(1+T4(x30))+-60.460720305222964\*ln(1+T5(x30))+24.392385930215696\*ln(1+T6(x30))+-5.664224581934092\*ln(1+T7(x30)))-1

psi231=exp(+-25833.867530853386\*ln(1+T0(x31))+29038.671597902634\*ln(1+T1(x31))+6332.200919636019\*ln(1+T2(x31))+25354.39786279269\*ln(1+T3(x31))+41602.64696883924\*ln(1+T4(x31))+55185.93700223231\*ln(1+T5(x31))+-121160.46845459547\*ln(1+T6(x31))+41761.987876771556\*ln(1+T7(x31)))-1

psi232=exp(+-25833.867530853386\*ln(1+T0(x32))+-55.80082418942823\*ln(1+T1(x32))+84.37588939182781\*ln(1+T2(x32))+-121.93863182380026\*ln(1+T3(x32))+143.55136438667435\*ln(1+T4(x32))+-50.78158690533036\*ln(1+T5(x32))+-21.960504894919787\*ln(1+T6(x32))+11.560107140046457\*ln(1+T7(x32)))-1

psi310=exp(+0.5013036320781262\*ln(1+T0(x10))+-1.1993148814149484\*ln(1+T1(x10))+0.6820414246032951\*ln(1+T2(x10))+0.7557221639986698\*ln(1+T3(x10))+-1.4083850032305554\*ln(1+T4(x10))+1.0926730142566987\*ln(1+T5(x10))+-0.021433195801318222\*ln(1+T6(x10))+-0.4763451065479972\*ln(1+T7(x10)))-1

psi311=exp(+0.5013036320781262\*ln(1+T0(x11))+0.9011367952928112\*ln(1+T1(x11))+-0.6162517239167409\*ln(1+T2(x11))+0.5933532054104036\*ln(1+T3(x11))+-0.2259693959288739\*ln(1+T4(x11))+-0.46541302764097175\*ln(1+T5(x11))+0.8512787199420359\*ln(1+T6(x11))+-0.4699642726293586\*ln(1+T7(x11)))-1

psi320=exp(+-0.4376029668301781\*ln(1+T0(x20))+2.902313191881633\*ln(1+T1(x20))+-1.650445776802811\*ln(1+T2(x20))+0.6226351100385576\*ln(1+T3(x20))+0.06343957522983587\*ln(1+T4(x20))+-0.17870016862918073\*ln(1+T5(x20))+-0.04512947910737732\*ln(1+T6(x20))+0.24418663183095585\*ln(1+T7(x20)))-1

psi321=exp(+-0.4376029668301781\*ln(1+T0(x21))+-0.5973704372369628\*ln(1+T1(x21))+-0.2131134223542977\*ln(1+T2(x21))+1.4823082720665863\*ln(1+T3(x21))+-2.555239521299849\*ln(1+T4(x21))+1.8385433343457125\*ln(1+T5(x21))+-0.27819544649947503\*ln(1+T6(x21))+-0.2290350026871884\*ln(1+T7(x21)))-1

psi330=exp(+-16493.7826773181\*ln(1+T0(x30))+-847.4634893092713\*ln(1+T1(x30))+434.5779311194289\*ln(1+T2(x30))+-147.17500580180192\*ln(1+T3(x30))+66.38792222512932\*ln(1+T4(x30))+-18.451588279250622\*ln(1+T5(x30))+-0.06502340462225535\*ln(1+T6(x30))+1.665194602873132\*ln(1+T7(x30)))-1

psi331=exp(+-16493.7826773181\*ln(1+T0(x31))+17779.324865357547\*ln(1+T1(x31))+-1382.0782791597017\*ln(1+T2(x31))+16625.628685928757\*ln(1+T3(x31))+26915.995021802457\*ln(1+T4(x31))+33026.67928785018\*ln(1+T5(x31))+-67150.70683580339\*ln(1+T6(x31))+24127.73717037718\*ln(1+T7(x31)))-1

psi332=exp(+-16493.7826773181\*ln(1+T0(x32))+-59.36741120210075\*ln(1+T1(x32))+-20.764136261239766\*ln(1+T2(x32))+107.45092711475306\*ln(1+T3(x32))+-182.53237074801612\*ln(1+T4(x32))+111.92082798786214\*ln(1+T5(x32))+-11.619605424024886\*ln(1+T6(x32))+-4.805404750128896\*ln(1+T7(x32)))-1

Ф01=+exp(+0.999979985069809\*ln(1+exp(+1.0078765932605016\*ln(1+T0(x10))+-9.375095617830254\*ln(1+T1(x10))+4.437805621241115\*ln(1+T2(x10))+-1.2691495973994007\*ln(1+T3(x10))+1.1966761918721602\*ln(1+T4(x10))+-1.5341251988476605\*ln(1+T5(x10))+1.237255058372709\*ln(1+T6(x10))+-0.4368878758516289\*ln(1+T7(x10)))-1)+0.9999804546432757\*ln(1+exp(+1.0078765932605016\*ln(1+T0(x11))+5.17384452598994\*ln(1+T1(x11))+-1.819836935542979\*ln(1+T2(x11))+-1.4549071149564028\*ln(1+T3(x11))+2.9345179308586515\*ln(1+T4(x11))+-1.6642417955025495\*ln(1+T5(x11))+-0.940641848314842\*ln(1+T6(x11))+1.197747428461699\*ln(1+T7(x11)))-1))-1

Ф02=+exp(+0.997003394763612\*ln(1+exp(+-0.25380979664043984\*ln(1+T0(x20))+0.4289045094610338\*ln(1+T1(x20))+0.7227283502179692\*ln(1+T2(x20))+-1.2989615969240558\*ln(1+T3(x20))+0.9008468089018122\*ln(1+T4(x20))+-0.3117120624809128\*ln(1+T5(x20))+-0.7341894036978045\*ln(1+T6(x20))+0.707519643773089\*ln(1+T7(x20)))-1)+1.0058817348244529\*ln(1+exp(+-0.25380979664043984\*ln(1+T0(x21))+-0.04675973354969946\*ln(1+T1(x21))+0.5226773495857509\*ln(1+T2(x21))+-1.7188528534431664\*ln(1+T3(x21))+2.0248058364011423\*ln(1+T4(x21))+-1.0495094035322556\*ln(1+T5(x21))+-0.16785532583648377\*ln(1+T6(x21))+0.20500566867309422\*ln(1+T7(x21)))-1))-1

Ф03=+exp(+0.9990626386638103\*ln(1+exp(+86423.03413424565\*ln(1+T0(x30))+-568.4563489162756\*ln(1+T1(x30))+304.01062413800054\*ln(1+T2(x30))+-68.39947832130575\*ln(1+T3(x30))+6.133133195584433\*ln(1+T4(x30))+0.6469420412788801\*ln(1+T5(x30))+-0.4908310171850466\*ln(1+T6(x30))+0.2660481857051654\*ln(1+T7(x30)))-1)+0.9990700421380578\*ln(1+exp(+86423.03413424565\*ln(1+T0(x31))+-86341.27009906256\*ln(1+T1(x31))+266439.6092565686\*ln(1+T2(x31))+-73379.06636020869\*ln(1+T3(x31))+-115616.57812721531\*ln(1+T4(x31))+-80417.9218773619\*ln(1+T5(x31))+-112044.95377087317\*ln(1+T6(x31))+-57655.34547189864\*ln(1+T7(x31)))-1)+0.9990774136798284\*ln(1+exp(+86423.03413424565\*ln(1+T0(x32))+154.2937517115147\*ln(1+T1(x32))+-94.3141571903461\*ln(1+T2(x32))+92.81059234963178\*ln(1+T3(x32))+-73.60407328539421\*ln(1+T4(x32))+-10.27566749418246\*ln(1+T5(x32))+24.46216126980048\*ln(1+T6(x32))+-1.977703102877775\*ln(1+T7(x32)))-1))-1

Ф11=+exp(+1.0000036243211117\*ln(1+exp(+3.3344016821136573\*ln(1+T0(x10))+9.461107273638747\*ln(1+T1(x10))+-5.176099921515887\*ln(1+T2(x10))+2.5498220661342472\*ln(1+T3(x10))+-2.0354403894204265\*ln(1+T4(x10))+1.260951903109428\*ln(1+T5(x10))+-0.664840028133066\*ln(1+T6(x10))+0.06851358705174758\*ln(1+T7(x10)))-1)+1.0000037943192137\*ln(1+exp(+3.3344016821136573\*ln(1+T0(x11))+-17.816737303546205\*ln(1+T1(x11))+7.517411166837675\*ln(1+T2(x11))+0.6224149969389731\*ln(1+T3(x11))+-0.9352015553388657\*ln(1+T4(x11))+-3.0918415974612015\*ln(1+T5(x11))+6.2628474783546775\*ln(1+T6(x11))+-3.72304083703611\*ln(1+T7(x11)))-1))-1

Ф12=+exp(+1.0030160956792877\*ln(1+exp(+0.12074874102958975\*ln(1+T0(x20))+0.012912051583008191\*ln(1+T1(x20))+-0.10779750487948979\*ln(1+T2(x20))+0.06551890720373688\*ln(1+T3(x20))+0.028817604548443364\*ln(1+T4(x20))+0.04504370027888293\*ln(1+T5(x20))+-0.033977997794590216\*ln(1+T6(x20))+-0.12144092008463372\*ln(1+T7(x20)))-1)+0.9982462993895251\*ln(1+exp(+0.12074874102958975\*ln(1+T0(x21))+0.07837474533891126\*ln(1+T1(x21))+-0.03462132190849608\*ln(1+T2(x21))+-0.11493689998702729\*ln(1+T3(x21))+-0.17955323271276938\*ln(1+T4(x21))+-0.02396041907274321\*ln(1+T5(x21))+0.06345518029529648\*ln(1+T6(x21))+0.10100055464722714\*ln(1+T7(x21)))-1))-1

Ф13=+exp(+0.9938592817991319\*ln(1+exp(+-1.5975054115402938\*ln(1+T0(x30))+4.762336376385421\*ln(1+T1(x30))+8.386999722312005\*ln(1+T2(x30))+-11.00044597032626\*ln(1+T3(x30))+0.20550284631159127\*ln(1+T4(x30))+10.584559205144119\*ln(1+T5(x30))+-13.70780546917582\*ln(1+T6(x30))+7.047799857475385\*ln(1+T7(x30)))-1)+0.9933649320871126\*ln(1+exp(+-1.5975054115402938\*ln(1+T0(x31))+6.464971342014239\*ln(1+T1(x31))+2.980242231078077\*ln(1+T2(x31))+-8.035874269878185\*ln(1+T3(x31))+-0.4022768792502549\*ln(1+T4(x31))+11.522586572836381\*ln(1+T5(x31))+2.0146703635742957\*ln(1+T6(x31))+-12.03635048231135\*ln(1+T7(x31)))-1)+0.9936458608099656\*ln(1+exp(+-1.5975054115402938\*ln(1+T0(x32))+-3.9039023114795564\*ln(1+T1(x32))+-2.100995583127316\*ln(1+T2(x32))+7.727994482336271\*ln(1+T3(x32))+7.128648036353176\*ln(1+T4(x32))+1.6304309695128028\*ln(1+T5(x32))+4.186026034933914\*ln(1+T6(x32))+-7.516694389282384\*ln(1+T7(x32)))-1))-1

Ф21=+exp(+0.999798919254976\*ln(1+exp(+-0.22056934268672942\*ln(1+T0(x10))+-1.3078134486919297\*ln(1+T1(x10))+0.34944768454305347\*ln(1+T2(x10))+1.2627586180426957\*ln(1+T3(x10))+-1.5936442015625891\*ln(1+T4(x10))+0.9505741542115419\*ln(1+T5(x10))+0.6797153789697775\*ln(1+T6(x10))+-0.9827966299625795\*ln(1+T7(x10)))-1)+0.9998898135530351\*ln(1+exp(+-0.22056934268672942\*ln(1+T0(x11))+3.4558082132958186\*ln(1+T1(x11))+-1.9181805223484345\*ln(1+T2(x11))+0.9682600258074707\*ln(1+T3(x11))+-0.5263606700339348\*ln(1+T4(x11))+-0.020690208490231236\*ln(1+T5(x11))+0.44779865216838816\*ln(1+T6(x11))+-0.309610313560924\*ln(1+T7(x11)))-1))-1

Ф22=+exp(+0.9999720214150322\*ln(1+exp(+-0.37845985682442645\*ln(1+T0(x20))+2.9097907299936243\*ln(1+T1(x20))+-1.504532983163441\*ln(1+T2(x20))+0.41364836780824965\*ln(1+T3(x20))+-0.2116645079264474\*ln(1+T4(x20))+0.7765563448716019\*ln(1+T5(x20))+-1.113143121489567\*ln(1+T6(x20))+0.7292271869541748\*ln(1+T7(x20)))-1)+0.9999959833796358\*ln(1+exp(+-0.37845985682442645\*ln(1+T0(x21))+-1.0748385375915812\*ln(1+T1(x21))+0.27361361685452656\*ln(1+T2(x21))+0.9340071396232857\*ln(1+T3(x21))+-1.8212287016937194\*ln(1+T4(x21))+1.442358704499684\*ln(1+T5(x21))+-0.3030399421439247\*ln(1+T6(x21))+-0.06306147162419234\*ln(1+T7(x21)))-1))-1

Ф23=+exp(+0.9946330737520668\*ln(1+exp(+-25833.867530853386\*ln(1+T0(x30))+-1276.7928585315635\*ln(1+T1(x30))+648.4858059614834\*ln(1+T2(x30))+-220.09014215052113\*ln(1+T3(x30))+119.89947402400779\*ln(1+T4(x30))+-60.460720305222964\*ln(1+T5(x30))+24.392385930215696\*ln(1+T6(x30))+-5.664224581934092\*ln(1+T7(x30)))-1)+0.9946429906133548\*ln(1+exp(+-25833.867530853386\*ln(1+T0(x31))+29038.671597902634\*ln(1+T1(x31))+6332.200919636019\*ln(1+T2(x31))+25354.39786279269\*ln(1+T3(x31))+41602.64696883924\*ln(1+T4(x31))+55185.93700223231\*ln(1+T5(x31))+-121160.46845459547\*ln(1+T6(x31))+41761.987876771556\*ln(1+T7(x31)))-1)+0.9946531491904997\*ln(1+exp(+-25833.867530853386\*ln(1+T0(x32))+-55.80082418942823\*ln(1+T1(x32))+84.37588939182781\*ln(1+T2(x32))+-121.93863182380026\*ln(1+T3(x32))+143.55136438667435\*ln(1+T4(x32))+-50.78158690533036\*ln(1+T5(x32))+-21.960504894919787\*ln(1+T6(x32))+11.560107140046457\*ln(1+T7(x32)))-1))-1

Ф31=+exp(+1.006004772751687\*ln(1+exp(+0.5013036320781262\*ln(1+T0(x10))+-1.1993148814149484\*ln(1+T1(x10))+0.6820414246032951\*ln(1+T2(x10))+0.7557221639986698\*ln(1+T3(x10))+-1.4083850032305554\*ln(1+T4(x10))+1.0926730142566987\*ln(1+T5(x10))+-0.021433195801318222\*ln(1+T6(x10))+-0.4763451065479972\*ln(1+T7(x10)))-1)+1.0031690038158885\*ln(1+exp(+0.5013036320781262\*ln(1+T0(x11))+0.9011367952928112\*ln(1+T1(x11))+-0.6162517239167409\*ln(1+T2(x11))+0.5933532054104036\*ln(1+T3(x11))+-0.2259693959288739\*ln(1+T4(x11))+-0.46541302764097175\*ln(1+T5(x11))+0.8512787199420359\*ln(1+T6(x11))+-0.4699642726293586\*ln(1+T7(x11)))-1))-1

Ф32=+exp(+0.9997697562849927\*ln(1+exp(+-0.4376029668301781\*ln(1+T0(x20))+2.902313191881633\*ln(1+T1(x20))+-1.650445776802811\*ln(1+T2(x20))+0.6226351100385576\*ln(1+T3(x20))+0.06343957522983587\*ln(1+T4(x20))+-0.17870016862918073\*ln(1+T5(x20))+-0.04512947910737732\*ln(1+T6(x20))+0.24418663183095585\*ln(1+T7(x20)))-1)+0.999661413055668\*ln(1+exp(+-0.4376029668301781\*ln(1+T0(x21))+-0.5973704372369628\*ln(1+T1(x21))+-0.2131134223542977\*ln(1+T2(x21))+1.4823082720665863\*ln(1+T3(x21))+-2.555239521299849\*ln(1+T4(x21))+1.8385433343457125\*ln(1+T5(x21))+-0.27819544649947503\*ln(1+T6(x21))+-0.2290350026871884\*ln(1+T7(x21)))-1))-1

Ф33=+exp(+0.9998515838531119\*ln(1+exp(+-16493.7826773181\*ln(1+T0(x30))+-847.4634893092713\*ln(1+T1(x30))+434.5779311194289\*ln(1+T2(x30))+-147.17500580180192\*ln(1+T3(x30))+66.38792222512932\*ln(1+T4(x30))+-18.451588279250622\*ln(1+T5(x30))+-0.06502340462225535\*ln(1+T6(x30))+1.665194602873132\*ln(1+T7(x30)))-1)+0.9998528759019447\*ln(1+exp(+-16493.7826773181\*ln(1+T0(x31))+17779.324865357547\*ln(1+T1(x31))+-1382.0782791597017\*ln(1+T2(x31))+16625.628685928757\*ln(1+T3(x31))+26915.995021802457\*ln(1+T4(x31))+33026.67928785018\*ln(1+T5(x31))+-67150.70683580339\*ln(1+T6(x31))+24127.73717037718\*ln(1+T7(x31)))-1)+0.9998542020188722\*ln(1+exp(+-16493.7826773181\*ln(1+T0(x32))+-59.36741120210075\*ln(1+T1(x32))+-20.764136261239766\*ln(1+T2(x32))+107.45092711475306\*ln(1+T3(x32))+-182.53237074801612\*ln(1+T4(x32))+111.92082798786214\*ln(1+T5(x32))+-11.619605424024886\*ln(1+T6(x32))+-4.805404750128896\*ln(1+T7(x32)))-1))-1

Ф0=exp(+0.2777925037212351\*ln(1++exp(+0.999979985069809\*ln(1+exp(+1.0078765932605016\*ln(1+T0(x10))+-9.375095617830254\*ln(1+T1(x10))+4.437805621241115\*ln(1+T2(x10))+-1.2691495973994007\*ln(1+T3(x10))+1.1966761918721602\*ln(1+T4(x10))+-1.5341251988476605\*ln(1+T5(x10))+1.237255058372709\*ln(1+T6(x10))+-0.4368878758516289\*ln(1+T7(x10)))-1)+0.9999804546432757\*ln(1+exp(+1.0078765932605016\*ln(1+T0(x11))+5.17384452598994\*ln(1+T1(x11))+-1.819836935542979\*ln(1+T2(x11))+-1.4549071149564028\*ln(1+T3(x11))+2.9345179308586515\*ln(1+T4(x11))+-1.6642417955025495\*ln(1+T5(x11))+-0.940641848314842\*ln(1+T6(x11))+1.197747428461699\*ln(1+T7(x11)))-1))-1)+-0.017697498807530437\*ln(1++exp(+0.997003394763612\*ln(1+exp(+-0.25380979664043984\*ln(1+T0(x20))+0.4289045094610338\*ln(1+T1(x20))+0.7227283502179692\*ln(1+T2(x20))+-1.2989615969240558\*ln(1+T3(x20))+0.9008468089018122\*ln(1+T4(x20))+-0.3117120624809128\*ln(1+T5(x20))+-0.7341894036978045\*ln(1+T6(x20))+0.707519643773089\*ln(1+T7(x20)))-1)+1.0058817348244529\*ln(1+exp(+-0.25380979664043984\*ln(1+T0(x21))+-0.04675973354969946\*ln(1+T1(x21))+0.5226773495857509\*ln(1+T2(x21))+-1.7188528534431664\*ln(1+T3(x21))+2.0248058364011423\*ln(1+T4(x21))+-1.0495094035322556\*ln(1+T5(x21))+-0.16785532583648377\*ln(1+T6(x21))+0.20500566867309422\*ln(1+T7(x21)))-1))-1)+0.7473137176032929\*ln(1++exp(+0.9990626386638103\*ln(1+exp(+86423.03413424565\*ln(1+T0(x30))+-568.4563489162756\*ln(1+T1(x30))+304.01062413800054\*ln(1+T2(x30))+-68.39947832130575\*ln(1+T3(x30))+6.133133195584433\*ln(1+T4(x30))+0.6469420412788801\*ln(1+T5(x30))+-0.4908310171850466\*ln(1+T6(x30))+0.2660481857051654\*ln(1+T7(x30)))-1)+0.9990700421380578\*ln(1+exp(+86423.03413424565\*ln(1+T0(x31))+-86341.27009906256\*ln(1+T1(x31))+266439.6092565686\*ln(1+T2(x31))+-73379.06636020869\*ln(1+T3(x31))+-115616.57812721531\*ln(1+T4(x31))+-80417.9218773619\*ln(1+T5(x31))+-112044.95377087317\*ln(1+T6(x31))+-57655.34547189864\*ln(1+T7(x31)))-1)+0.9990774136798284\*ln(1+exp(+86423.03413424565\*ln(1+T0(x32))+154.2937517115147\*ln(1+T1(x32))+-94.3141571903461\*ln(1+T2(x32))+92.81059234963178\*ln(1+T3(x32))+-73.60407328539421\*ln(1+T4(x32))+-10.27566749418246\*ln(1+T5(x32))+24.46216126980048\*ln(1+T6(x32))+-1.977703102877775\*ln(1+T7(x32)))-1))-1))-1

Ф1=exp(+0.2651890696553106\*ln(1++exp(+1.0000036243211117\*ln(1+exp(+3.3344016821136573\*ln(1+T0(x10))+9.461107273638747\*ln(1+T1(x10))+-5.176099921515887\*ln(1+T2(x10))+2.5498220661342472\*ln(1+T3(x10))+-2.0354403894204265\*ln(1+T4(x10))+1.260951903109428\*ln(1+T5(x10))+-0.664840028133066\*ln(1+T6(x10))+0.06851358705174758\*ln(1+T7(x10)))-1)+1.0000037943192137\*ln(1+exp(+3.3344016821136573\*ln(1+T0(x11))+-17.816737303546205\*ln(1+T1(x11))+7.517411166837675\*ln(1+T2(x11))+0.6224149969389731\*ln(1+T3(x11))+-0.9352015553388657\*ln(1+T4(x11))+-3.0918415974612015\*ln(1+T5(x11))+6.2628474783546775\*ln(1+T6(x11))+-3.72304083703611\*ln(1+T7(x11)))-1))-1)+0.016192961178925276\*ln(1++exp(+1.0030160956792877\*ln(1+exp(+0.12074874102958975\*ln(1+T0(x20))+0.012912051583008191\*ln(1+T1(x20))+-0.10779750487948979\*ln(1+T2(x20))+0.06551890720373688\*ln(1+T3(x20))+0.028817604548443364\*ln(1+T4(x20))+0.04504370027888293\*ln(1+T5(x20))+-0.033977997794590216\*ln(1+T6(x20))+-0.12144092008463372\*ln(1+T7(x20)))-1)+0.9982462993895251\*ln(1+exp(+0.12074874102958975\*ln(1+T0(x21))+0.07837474533891126\*ln(1+T1(x21))+-0.03462132190849608\*ln(1+T2(x21))+-0.11493689998702729\*ln(1+T3(x21))+-0.17955323271276938\*ln(1+T4(x21))+-0.02396041907274321\*ln(1+T5(x21))+0.06345518029529648\*ln(1+T6(x21))+0.10100055464722714\*ln(1+T7(x21)))-1))-1)+0.7409149993772992\*ln(1++exp(+0.9938592817991319\*ln(1+exp(+-1.5975054115402938\*ln(1+T0(x30))+4.762336376385421\*ln(1+T1(x30))+8.386999722312005\*ln(1+T2(x30))+-11.00044597032626\*ln(1+T3(x30))+0.20550284631159127\*ln(1+T4(x30))+10.584559205144119\*ln(1+T5(x30))+-13.70780546917582\*ln(1+T6(x30))+7.047799857475385\*ln(1+T7(x30)))-1)+0.9933649320871126\*ln(1+exp(+-1.5975054115402938\*ln(1+T0(x31))+6.464971342014239\*ln(1+T1(x31))+2.980242231078077\*ln(1+T2(x31))+-8.035874269878185\*ln(1+T3(x31))+-0.4022768792502549\*ln(1+T4(x31))+11.522586572836381\*ln(1+T5(x31))+2.0146703635742957\*ln(1+T6(x31))+-12.03635048231135\*ln(1+T7(x31)))-1)+0.9936458608099656\*ln(1+exp(+-1.5975054115402938\*ln(1+T0(x32))+-3.9039023114795564\*ln(1+T1(x32))+-2.100995583127316\*ln(1+T2(x32))+7.727994482336271\*ln(1+T3(x32))+7.128648036353176\*ln(1+T4(x32))+1.6304309695128028\*ln(1+T5(x32))+4.186026034933914\*ln(1+T6(x32))+-7.516694389282384\*ln(1+T7(x32)))-1))-1))-1

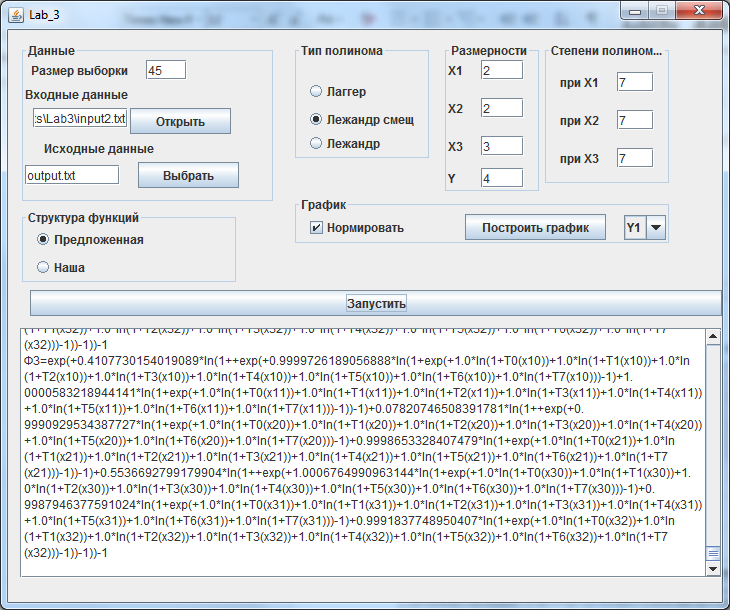
Ф2=exp(+0.1213056516002732\*ln(1++exp(+0.999798919254976\*ln(1+exp(+-0.22056934268672942\*ln(1+T0(x10))+-1.3078134486919297\*ln(1+T1(x10))+0.34944768454305347\*ln(1+T2(x10))+1.2627586180426957\*ln(1+T3(x10))+-1.5936442015625891\*ln(1+T4(x10))+0.9505741542115419\*ln(1+T5(x10))+0.6797153789697775\*ln(1+T6(x10))+-0.9827966299625795\*ln(1+T7(x10)))-1)+0.9998898135530351\*ln(1+exp(+-0.22056934268672942\*ln(1+T0(x11))+3.4558082132958186\*ln(1+T1(x11))+-1.9181805223484345\*ln(1+T2(x11))+0.9682600258074707\*ln(1+T3(x11))+-0.5263606700339348\*ln(1+T4(x11))+-0.020690208490231236\*ln(1+T5(x11))+0.44779865216838816\*ln(1+T6(x11))+-0.309610313560924\*ln(1+T7(x11)))-1))-1)+0.008440332996530672\*ln(1++exp(+0.9999720214150322\*ln(1+exp(+-0.37845985682442645\*ln(1+T0(x20))+2.9097907299936243\*ln(1+T1(x20))+-1.504532983163441\*ln(1+T2(x20))+0.41364836780824965\*ln(1+T3(x20))+-0.2116645079264474\*ln(1+T4(x20))+0.7765563448716019\*ln(1+T5(x20))+-1.113143121489567\*ln(1+T6(x20))+0.7292271869541748\*ln(1+T7(x20)))-1)+0.9999959833796358\*ln(1+exp(+-0.37845985682442645\*ln(1+T0(x21))+-1.0748385375915812\*ln(1+T1(x21))+0.27361361685452656\*ln(1+T2(x21))+0.9340071396232857\*ln(1+T3(x21))+-1.8212287016937194\*ln(1+T4(x21))+1.442358704499684\*ln(1+T5(x21))+-0.3030399421439247\*ln(1+T6(x21))+-0.06306147162419234\*ln(1+T7(x21)))-1))-1)+0.8857109585651791\*ln(1++exp(+0.9946330737520668\*ln(1+exp(+-25833.867530853386\*ln(1+T0(x30))+-1276.7928585315635\*ln(1+T1(x30))+648.4858059614834\*ln(1+T2(x30))+-220.09014215052113\*ln(1+T3(x30))+119.89947402400779\*ln(1+T4(x30))+-60.460720305222964\*ln(1+T5(x30))+24.392385930215696\*ln(1+T6(x30))+-5.664224581934092\*ln(1+T7(x30)))-1)+0.9946429906133548\*ln(1+exp(+-25833.867530853386\*ln(1+T0(x31))+29038.671597902634\*ln(1+T1(x31))+6332.200919636019\*ln(1+T2(x31))+25354.39786279269\*ln(1+T3(x31))+41602.64696883924\*ln(1+T4(x31))+55185.93700223231\*ln(1+T5(x31))+-121160.46845459547\*ln(1+T6(x31))+41761.987876771556\*ln(1+T7(x31)))-1)+0.9946531491904997\*ln(1+exp(+-25833.867530853386\*ln(1+T0(x32))+-55.80082418942823\*ln(1+T1(x32))+84.37588939182781\*ln(1+T2(x32))+-121.93863182380026\*ln(1+T3(x32))+143.55136438667435\*ln(1+T4(x32))+-50.78158690533036\*ln(1+T5(x32))+-21.960504894919787\*ln(1+T6(x32))+11.560107140046457\*ln(1+T7(x32)))-1))-1))-1

Ф3=exp(+0.21015226206741255\*ln(1++exp(+1.006004772751687\*ln(1+exp(+0.5013036320781262\*ln(1+T0(x10))+-1.1993148814149484\*ln(1+T1(x10))+0.6820414246032951\*ln(1+T2(x10))+0.7557221639986698\*ln(1+T3(x10))+-1.4083850032305554\*ln(1+T4(x10))+1.0926730142566987\*ln(1+T5(x10))+-0.021433195801318222\*ln(1+T6(x10))+-0.4763451065479972\*ln(1+T7(x10)))-1)+1.0031690038158885\*ln(1+exp(+0.5013036320781262\*ln(1+T0(x11))+0.9011367952928112\*ln(1+T1(x11))+-0.6162517239167409\*ln(1+T2(x11))+0.5933532054104036\*ln(1+T3(x11))+-0.2259693959288739\*ln(1+T4(x11))+-0.46541302764097175\*ln(1+T5(x11))+0.8512787199420359\*ln(1+T6(x11))+-0.4699642726293586\*ln(1+T7(x11)))-1))-1)+0.03706438576833786\*ln(1++exp(+0.9997697562849927\*ln(1+exp(+-0.4376029668301781\*ln(1+T0(x20))+2.902313191881633\*ln(1+T1(x20))+-1.650445776802811\*ln(1+T2(x20))+0.6226351100385576\*ln(1+T3(x20))+0.06343957522983587\*ln(1+T4(x20))+-0.17870016862918073\*ln(1+T5(x20))+-0.04512947910737732\*ln(1+T6(x20))+0.24418663183095585\*ln(1+T7(x20)))-1)+0.999661413055668\*ln(1+exp(+-0.4376029668301781\*ln(1+T0(x21))+-0.5973704372369628\*ln(1+T1(x21))+-0.2131134223542977\*ln(1+T2(x21))+1.4823082720665863\*ln(1+T3(x21))+-2.555239521299849\*ln(1+T4(x21))+1.8385433343457125\*ln(1+T5(x21))+-0.27819544649947503\*ln(1+T6(x21))+-0.2290350026871884\*ln(1+T7(x21)))-1))-1)+0.7747263943078648\*ln(1++exp(+0.9998515838531119\*ln(1+exp(+-16493.7826773181\*ln(1+T0(x30))+-847.4634893092713\*ln(1+T1(x30))+434.5779311194289\*ln(1+T2(x30))+-147.17500580180192\*ln(1+T3(x30))+66.38792222512932\*ln(1+T4(x30))+-18.451588279250622\*ln(1+T5(x30))+-0.06502340462225535\*ln(1+T6(x30))+1.665194602873132\*ln(1+T7(x30)))-1)+0.9998528759019447\*ln(1+exp(+-16493.7826773181\*ln(1+T0(x31))+17779.324865357547\*ln(1+T1(x31))+-1382.0782791597017\*ln(1+T2(x31))+16625.628685928757\*ln(1+T3(x31))+26915.995021802457\*ln(1+T4(x31))+33026.67928785018\*ln(1+T5(x31))+-67150.70683580339\*ln(1+T6(x31))+24127.73717037718\*ln(1+T7(x31)))-1)+0.9998542020188722\*ln(1+exp(+-16493.7826773181\*ln(1+T0(x32))+-59.36741120210075\*ln(1+T1(x32))+-20.764136261239766\*ln(1+T2(x32))+107.45092711475306\*ln(1+T3(x32))+-182.53237074801612\*ln(1+T4(x32))+111.92082798786214\*ln(1+T5(x32))+-11.619605424024886\*ln(1+T6(x32))+-4.805404750128896\*ln(1+T7(x32)))-1))-1))-1

# *2 вид:* Смещенный полином Лежандра





  
  
psi010=exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1

psi011=exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1

psi020=exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1

psi021=exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1

psi030=exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1

psi031=exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1

psi032=exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1

psi110=exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1

psi111=exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1

psi120=exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1

psi121=exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1

psi130=exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1

psi131=exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1

psi132=exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1

psi210=exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1

psi211=exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1

psi220=exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1

psi221=exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1

psi230=exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1

psi231=exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1

psi232=exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1

psi310=exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1

psi311=exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1

psi320=exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1

psi321=exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1

psi330=exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1

psi331=exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1

psi332=exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1

Ф01=+exp(+0.9999886708694129\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1)+0.9998451433524178\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1))-1

Ф02=+exp(+1.013841361953102\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1)+0.9975499380968393\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1))-1

Ф03=+exp(+0.9918556990920453\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1)+0.9945560443575511\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1)+0.9972564233166274\*ln(1+exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1))-1

Ф11=+exp(+1.0191343693678914\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1)+0.9979211423662734\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1))-1

Ф12=+exp(+0.993053164793129\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1)+1.0009399835668045\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1))-1

Ф13=+exp(+0.9289255630139406\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1)+0.9272608266662953\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1)+0.925596551935113\*ln(1+exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1))-1

Ф21=+exp(+1.0006540354596032\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1)+0.9995546074480584\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1))-1

Ф22=+exp(+0.9997380293106897\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1)+1.000263564381721\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1))-1

Ф23=+exp(+0.9724091203807913\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1)+0.9720905901796262\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1)+0.9717722033600452\*ln(1+exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1))-1

Ф31=+exp(+0.9999726189056888\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1)+1.0000583218944141\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1))-1

Ф32=+exp(+0.9990929534387727\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1)+0.9998653328407479\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1))-1

Ф33=+exp(+1.0006764990963144\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1)+0.9987946377591024\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1)+0.9991837748950407\*ln(1+exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1))-1

Ф0=exp(+0.3288914646140879\*ln(1++exp(+0.9999886708694129\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1)+0.9998451433524178\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1))-1)+-0.020440592361916272\*ln(1++exp(+1.013841361953102\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1)+0.9975499380968393\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1))-1)+0.6943026537983414\*ln(1++exp(+0.9918556990920453\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1)+0.9945560443575511\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1)+0.9972564233166274\*ln(1+exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1))-1))-1

Ф1=exp(+0.215247468844396\*ln(1++exp(+1.0191343693678914\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1)+0.9979211423662734\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1))-1)+0.057047164862960194\*ln(1++exp(+0.993053164793129\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1)+1.0009399835668045\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1))-1)+0.7609470389512006\*ln(1++exp(+0.9289255630139406\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1)+0.9272608266662953\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1)+0.925596551935113\*ln(1+exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1))-1))-1

Ф2=exp(+0.27930790351566026\*ln(1++exp(+1.0006540354596032\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1)+0.9995546074480584\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1))-1)+0.03616331758162756\*ln(1++exp(+0.9997380293106897\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1)+1.000263564381721\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1))-1)+0.7255965982220096\*ln(1++exp(+0.9724091203807913\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1)+0.9720905901796262\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1)+0.9717722033600452\*ln(1+exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1))-1))-1

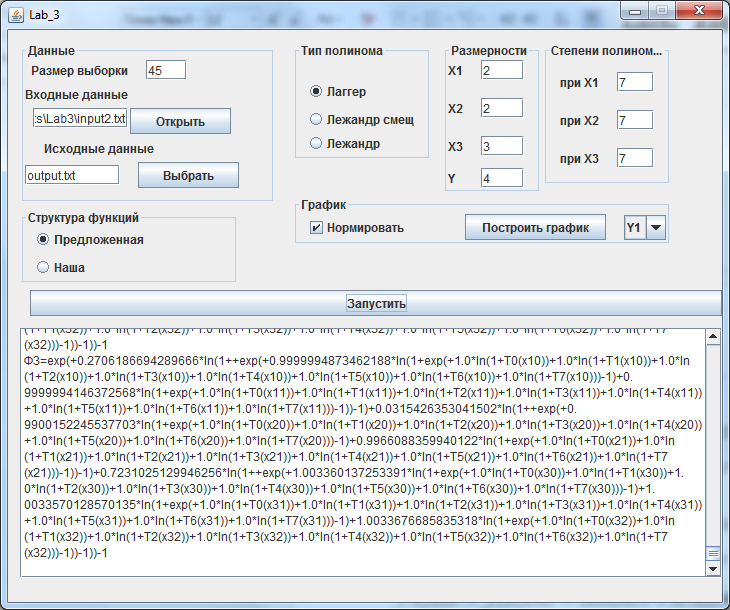
Ф3=exp(+0.4107730154019089\*ln(1++exp(+0.9999726189056888\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10)))-1)+1.0000583218944141\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11)))-1))-1)+0.07820746508391781\*ln(1++exp(+0.9990929534387727\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20)))-1)+0.9998653328407479\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21)))-1))-1)+0.5536692799179904\*ln(1++exp(+1.0006764990963144\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30)))-1)+0.9987946377591024\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31)))-1)+0.9991837748950407\*ln(1+exp(+1.0\*ln(1+T0(x32))+1.0\*ln(1+T1(x32))+1.0\*ln(1+T2(x32))+1.0\*ln(1+T3(x32))+1.0\*ln(1+T4(x32))+1.0\*ln(1+T5(x32))+1.0\*ln(1+T6(x32))+1.0\*ln(1+T7(x32)))-1))-1))-1

***3 вид:***  
Полином Лаггера



 но 

 ; 



psi010=exp(+3.838110893199928\*ln(1+T0(x10))+1.5309520249489794\*ln(1+T1(x10))+-0.7762068335261156\*ln(1+T2(x10))+0.6850741510062301\*ln(1+T3(x10))+-2.279384942782013\*ln(1+T4(x10))+4.242414283199751\*ln(1+T5(x10))+-4.292187363020475\*ln(1+T6(x10))+1.6997907701386648\*ln(1+T7(x10)))-1

psi011=exp(+3.838110893199928\*ln(1+T0(x11))+-6.3722266722308865\*ln(1+T1(x11))+-16.58256423172599\*ln(1+T2(x11))+18.29409835573527\*ln(1+T3(x11))+7.876481181841437\*ln(1+T4(x11))+-33.617619247378165\*ln(1+T5(x11))+35.8793044856896\*ln(1+T6(x11))+-14.0264752503876\*ln(1+T7(x11)))-1

psi020=exp(+45.58847865637187\*ln(1+T0(x20))+-50.617665821879115\*ln(1+T1(x20))+-146.82381026937128\*ln(1+T2(x20))+229.29248925498138\*ln(1+T3(x20))+-158.2215202518411\*ln(1+T4(x20))+100.97444892226179\*ln(1+T5(x20))+-51.22397100803834\*ln(1+T6(x20))+13.106055057733354\*ln(1+T7(x20)))-1

psi021=exp(+45.58847865637187\*ln(1+T0(x21))+-17.115340815417426\*ln(1+T1(x21))+-79.81916013139465\*ln(1+T2(x21))+120.30667148691144\*ln(1+T3(x21))+-84.60931969098375\*ln(1+T4(x21))+48.62148927926165\*ln(1+T5(x21))+-16.28706858417513\*ln(1+T6(x21))+1.125060524460816\*ln(1+T7(x21)))-1

psi030=exp(+378.0865952855692\*ln(1+T0(x30))+-17.371246467452302\*ln(1+T1(x30))+-412.3468065046327\*ln(1+T2(x30))+652.6643152540636\*ln(1+T3(x30))+-874.8162833229435\*ln(1+T4(x30))+1411.874938026344\*ln(1+T5(x30))+-1612.5545996701737\*ln(1+T6(x30))+737.4285877771393\*ln(1+T7(x30)))-1

psi031=exp(+378.0865952855692\*ln(1+T0(x31))+206.47620531632762\*ln(1+T1(x31))+35.11069644403024\*ln(1+T2(x31))+-75.77069683875955\*ln(1+T3(x31))+-77.32488086534003\*ln(1+T4(x31))+-29.49328702631603\*ln(1+T5(x31))+14.254514179546605\*ln(1+T6(x31))+41.83010199893179\*ln(1+T7(x31)))-1

psi032=exp(+378.0865952855692\*ln(1+T0(x32))+-1119.156263363238\*ln(1+T1(x32))+-2616.083116529668\*ln(1+T2(x32))+4976.854025396079\*ln(1+T3(x32))+-4977.018458694015\*ln(1+T4(x32))+4087.6303613512637\*ln(1+T5(x32))+-1978.4105977252473\*ln(1+T6(x32))+394.5335156222737\*ln(1+T7(x32)))-1

psi110=exp(+18.503035665678645\*ln(1+T0(x10))+-44.949936648926695\*ln(1+T1(x10))+-108.4029090923041\*ln(1+T2(x10))+181.24950965880515\*ln(1+T3(x10))+-137.33439102872822\*ln(1+T4(x10))+98.01502477631169\*ln(1+T5(x10))+-55.04210196521577\*ln(1+T6(x10))+15.795795596990004\*ln(1+T7(x10)))-1

psi111=exp(+18.503035665678645\*ln(1+T0(x11))+16.763806540738184\*ln(1+T1(x11))+15.024577330963218\*ln(1+T2(x11))+-24.35522022084423\*ln(1+T3(x11))+0.17902989842808203\*ln(1+T4(x11))+22.795743462926524\*ln(1+T5(x11))+-27.45624808903215\*ln(1+T6(x11))+11.424741650692706\*ln(1+T7(x11)))-1

psi120=exp(+0.09732534744617177\*ln(1+T0(x20))+0.0755888987815842\*ln(1+T1(x20))+0.05385245011699665\*ln(1+T2(x20))+0.03380192749854132\*ln(1+T3(x20))+0.017841308312352134\*ln(1+T4(x20))+0.006357033086470353\*ln(1+T5(x20))+-0.0019450927975522316\*ln(1+T6(x20))+-0.00803742238432353\*ln(1+T7(x20)))-1

psi121=exp(+0.09732534744617177\*ln(1+T0(x21))+0.07104173640455518\*ln(1+T1(x21))+0.04475812536293872\*ln(1+T2(x21))+0.017745361446817154\*ln(1+T3(x21))+-0.009770345874704147\*ln(1+T4(x21))+-0.03374017343582745\*ln(1+T5(x21))+-0.05118259391097622\*ln(1+T6(x21))+-0.06197732236587362\*ln(1+T7(x21)))-1

psi130=exp(+29.500854346985555\*ln(1+T0(x30))+-46.16824015385558\*ln(1+T1(x30))+-121.83733745999889\*ln(1+T2(x30))+176.18779905570915\*ln(1+T3(x30))+-47.58401583657103\*ln(1+T4(x30))+-123.79905651572749\*ln(1+T5(x30))+224.32895982520978\*ln(1+T6(x30))+-116.53386189013663\*ln(1+T7(x30)))-1

psi131=exp(+29.500854346985555\*ln(1+T0(x31))+28.064631829361407\*ln(1+T1(x31))+26.628417242848098\*ln(1+T2(x31))+20.180255851073184\*ln(1+T3(x31))+4.616164750587913\*ln(1+T4(x31))+-15.0322697915982\*ln(1+T5(x31))+-34.32185266920681\*ln(1+T6(x31))+-52.28811215385552\*ln(1+T7(x31)))-1

psi132=exp(+29.500854346985555\*ln(1+T0(x32))+-4.541621070374992\*ln(1+T1(x32))+-38.58409804044643\*ln(1+T2(x32))+108.7404978148402\*ln(1+T3(x32))+-122.15195798118881\*ln(1+T4(x32))+-74.46412706531274\*ln(1+T5(x32))+312.91779054234513\*ln(1+T6(x32))+-181.10355922606834\*ln(1+T7(x32)))-1

psi210=exp(+-2.181882861709858\*ln(1+T0(x10))+4.92115361201848\*ln(1+T1(x10))+12.024190085850053\*ln(1+T2(x10))+-11.557578947803304\*ln(1+T3(x10))+-7.305872068785254\*ln(1+T4(x10))+15.160109609230659\*ln(1+T5(x10))+-8.59664546795218\*ln(1+T6(x10))+1.4389062818158873\*ln(1+T7(x10)))-1

psi211=exp(+-2.181882861709858\*ln(1+T0(x11))+-1.9070699593984872\*ln(1+T1(x11))+-1.6322570573191686\*ln(1+T2(x11))+2.301881527374795\*ln(1+T3(x11))+1.0338143336905485\*ln(1+T4(x11))+-4.158625744992928\*ln(1+T5(x11))+4.452541534602822\*ln(1+T6(x11))+-1.737700875187219\*ln(1+T7(x11)))-1

psi220=exp(+7.777468214075455\*ln(1+T0(x20))+-36.44726602726387\*ln(1+T1(x20))+-80.6720002858276\*ln(1+T2(x20))+128.17178818167355\*ln(1+T3(x20))+-82.86046870624475\*ln(1+T4(x20))+49.24027092495042\*ln(1+T5(x20))+-23.46103739549334\*ln(1+T6(x20))+5.966520968914767\*ln(1+T7(x20)))-1

psi221=exp(+7.777468214075455\*ln(1+T0(x21))+25.846082234968463\*ln(1+T1(x21))+43.914699170018494\*ln(1+T2(x21))+-79.88061256335635\*ln(1+T3(x21))+59.74531652165636\*ln(1+T4(x21))+-35.39178807490148\*ln(1+T5(x21))+11.568416873589674\*ln(1+T6(x21))+-0.8091645263069945\*ln(1+T7(x21)))-1

psi230=exp(+143.85562255220825\*ln(1+T0(x30))+-81.77097707577096\*ln(1+T1(x30))+-307.2501393782645\*ln(1+T2(x30))+512.1522046382631\*ln(1+T3(x30))+-712.9534804378683\*ln(1+T4(x30))+1289.6282237186126\*ln(1+T5(x30))+-1644.5275356238267\*ln(1+T6(x30))+801.1708376692931\*ln(1+T7(x30)))-1

psi231=exp(+143.85562255220825\*ln(1+T0(x31))+67.70231821645648\*ln(1+T1(x31))+-8.394617444173624\*ln(1+T2(x31))+-54.01362159230008\*ln(1+T3(x31))+-44.46772853902575\*ln(1+T4(x31))+-10.095282749714658\*ln(1+T5(x31))+22.056251835539836\*ln(1+T6(x31))+46.017589157079776\*ln(1+T7(x31)))-1

psi232=exp(+143.85562255220825\*ln(1+T0(x32))+-369.41463853565114\*ln(1+T1(x32))+-882.5320054352666\*ln(1+T2(x32))+1569.5154740031605\*ln(1+T3(x32))+-1296.379151854587\*ln(1+T4(x32))+537.6340053510274\*ln(1+T5(x32))+69.92363493206844\*ln(1+T6(x32))+-70.22126890106111\*ln(1+T7(x32)))-1

psi310=exp(+-2.1733962322649654\*ln(1+T0(x10))+4.902191700861223\*ln(1+T1(x10))+11.977779634069579\*ln(1+T2(x10))+-11.83497902398336\*ln(1+T3(x10))+-6.912840937191477\*ln(1+T4(x10))+17.776334102863355\*ln(1+T5(x10))+-14.660493813597842\*ln(1+T6(x10))+4.757549766771358\*ln(1+T7(x10)))-1

psi311=exp(+-2.1733962322649654\*ln(1+T0(x11))+-1.8454198829663737\*ln(1+T1(x11))+-1.517443533750061\*ln(1+T2(x11))+2.0106817942693427\*ln(1+T3(x11))+1.1657677146900756\*ln(1+T4(x11))+-3.590912055801638\*ln(1+T5(x11))+3.2578788355532278\*ln(1+T6(x11))+-1.1069281586383286\*ln(1+T7(x11)))-1

psi320=exp(+0.8071305258059297\*ln(1+T0(x20))+-0.017017907042396228\*ln(1+T1(x20))+-0.8411663398899353\*ln(1+T2(x20))+0.17872238843278174\*ln(1+T3(x20))+1.0638737509039984\*ln(1+T4(x20))+-0.6460131101194501\*ln(1+T5(x20))+-0.5343571788922898\*ln(1+T6(x20))+0.680811251176263\*ln(1+T7(x20)))-1

psi321=exp(+0.8071305258059297\*ln(1+T0(x21))+-0.6234702951320609\*ln(1+T1(x21))+-2.0540711160739256\*ln(1+T2(x21))+0.0931162263847704\*ln(1+T3(x21))+3.7692677658267666\*ln(1+T4(x21))+-0.75359343398973\*ln(1+T5(x21))+-4.464557414773323\*ln(1+T6(x21))+2.8455267264680484\*ln(1+T7(x21)))-1

psi330=exp(+177.9605073711452\*ln(1+T0(x30))+-312.1281429691977\*ln(1+T1(x30))+-802.2148353194368\*ln(1+T2(x30))+1439.5662339374583\*ln(1+T3(x30))+-1553.930084848189\*ln(1+T4(x30))+1941.1101951985374\*ln(1+T5(x30))+-1850.513748167983\*ln(1+T6(x30))+764.566540619912\*ln(1+T7(x30)))-1

psi331=exp(+177.9605073711452\*ln(1+T0(x31))+101.51656809814367\*ln(1+T1(x31))+25.08428692033226\*ln(1+T2(x31))+-27.115859990943296\*ln(1+T3(x31))+-35.28622622387218\*ln(1+T4(x31))+-23.743626539758342\*ln(1+T5(x31))+-13.916614201306235\*ln(1+T6(x31))+-10.498974098454187\*ln(1+T7(x31)))-1

psi332=exp(+177.9605073711452\*ln(1+T0(x32))+-194.25740830485233\*ln(1+T1(x32))+-566.4725605085316\*ln(1+T2(x32))+975.7116893339941\*ln(1+T3(x32))+-785.6278408197325\*ln(1+T4(x32))+256.2145117873805\*ln(1+T5(x32))+146.48769364706817\*ln(1+T6(x32))+-88.0622411749155\*ln(1+T7(x32)))-1

Ф01=+exp(+1.0002347156990836\*ln(1+exp(+3.838110893199928\*ln(1+T0(x10))+1.5309520249489794\*ln(1+T1(x10))+-0.7762068335261156\*ln(1+T2(x10))+0.6850741510062301\*ln(1+T3(x10))+-2.279384942782013\*ln(1+T4(x10))+4.242414283199751\*ln(1+T5(x10))+-4.292187363020475\*ln(1+T6(x10))+1.6997907701386648\*ln(1+T7(x10)))-1)+1.0002450026563976\*ln(1+exp(+3.838110893199928\*ln(1+T0(x11))+-6.3722266722308865\*ln(1+T1(x11))+-16.58256423172599\*ln(1+T2(x11))+18.29409835573527\*ln(1+T3(x11))+7.876481181841437\*ln(1+T4(x11))+-33.617619247378165\*ln(1+T5(x11))+35.8793044856896\*ln(1+T6(x11))+-14.0264752503876\*ln(1+T7(x11)))-1))-1

Ф02=+exp(+0.9999949176853706\*ln(1+exp(+45.58847865637187\*ln(1+T0(x20))+-50.617665821879115\*ln(1+T1(x20))+-146.82381026937128\*ln(1+T2(x20))+229.29248925498138\*ln(1+T3(x20))+-158.2215202518411\*ln(1+T4(x20))+100.97444892226179\*ln(1+T5(x20))+-51.22397100803834\*ln(1+T6(x20))+13.106055057733354\*ln(1+T7(x20)))-1)+0.9999963471330582\*ln(1+exp(+45.58847865637187\*ln(1+T0(x21))+-17.115340815417426\*ln(1+T1(x21))+-79.81916013139465\*ln(1+T2(x21))+120.30667148691144\*ln(1+T3(x21))+-84.60931969098375\*ln(1+T4(x21))+48.62148927926165\*ln(1+T5(x21))+-16.28706858417513\*ln(1+T6(x21))+1.125060524460816\*ln(1+T7(x21)))-1))-1

Ф03=+exp(+1.0052204351945253\*ln(1+exp(+378.0865952855692\*ln(1+T0(x30))+-17.371246467452302\*ln(1+T1(x30))+-412.3468065046327\*ln(1+T2(x30))+652.6643152540636\*ln(1+T3(x30))+-874.8162833229435\*ln(1+T4(x30))+1411.874938026344\*ln(1+T5(x30))+-1612.5545996701737\*ln(1+T6(x30))+737.4285877771393\*ln(1+T7(x30)))-1)+1.0052311769464735\*ln(1+exp(+378.0865952855692\*ln(1+T0(x31))+206.47620531632762\*ln(1+T1(x31))+35.11069644403024\*ln(1+T2(x31))+-75.77069683875955\*ln(1+T3(x31))+-77.32488086534003\*ln(1+T4(x31))+-29.49328702631603\*ln(1+T5(x31))+14.254514179546605\*ln(1+T6(x31))+41.83010199893179\*ln(1+T7(x31)))-1)+1.0052273673030179\*ln(1+exp(+378.0865952855692\*ln(1+T0(x32))+-1119.156263363238\*ln(1+T1(x32))+-2616.083116529668\*ln(1+T2(x32))+4976.854025396079\*ln(1+T3(x32))+-4977.018458694015\*ln(1+T4(x32))+4087.6303613512637\*ln(1+T5(x32))+-1978.4105977252473\*ln(1+T6(x32))+394.5335156222737\*ln(1+T7(x32)))-1))-1

Ф11=+exp(+0.9869145648552234\*ln(1+exp(+18.503035665678645\*ln(1+T0(x10))+-44.949936648926695\*ln(1+T1(x10))+-108.4029090923041\*ln(1+T2(x10))+181.24950965880515\*ln(1+T3(x10))+-137.33439102872822\*ln(1+T4(x10))+98.01502477631169\*ln(1+T5(x10))+-55.04210196521577\*ln(1+T6(x10))+15.795795596990004\*ln(1+T7(x10)))-1)+0.987075190600403\*ln(1+exp(+18.503035665678645\*ln(1+T0(x11))+16.763806540738184\*ln(1+T1(x11))+15.024577330963218\*ln(1+T2(x11))+-24.35522022084423\*ln(1+T3(x11))+0.17902989842808203\*ln(1+T4(x11))+22.795743462926524\*ln(1+T5(x11))+-27.45624808903215\*ln(1+T6(x11))+11.424741650692706\*ln(1+T7(x11)))-1))-1

Ф12=+exp(+0.7740391863208671\*ln(1+exp(+0.09732534744617177\*ln(1+T0(x20))+0.0755888987815842\*ln(1+T1(x20))+0.05385245011699665\*ln(1+T2(x20))+0.03380192749854132\*ln(1+T3(x20))+0.017841308312352134\*ln(1+T4(x20))+0.006357033086470353\*ln(1+T5(x20))+-0.0019450927975522316\*ln(1+T6(x20))+-0.00803742238432353\*ln(1+T7(x20)))-1)+1.1566698616221043\*ln(1+exp(+0.09732534744617177\*ln(1+T0(x21))+0.07104173640455518\*ln(1+T1(x21))+0.04475812536293872\*ln(1+T2(x21))+0.017745361446817154\*ln(1+T3(x21))+-0.009770345874704147\*ln(1+T4(x21))+-0.03374017343582745\*ln(1+T5(x21))+-0.05118259391097622\*ln(1+T6(x21))+-0.06197732236587362\*ln(1+T7(x21)))-1))-1

Ф13=+exp(+1.000557858512941\*ln(1+exp(+29.500854346985555\*ln(1+T0(x30))+-46.16824015385558\*ln(1+T1(x30))+-121.83733745999889\*ln(1+T2(x30))+176.18779905570915\*ln(1+T3(x30))+-47.58401583657103\*ln(1+T4(x30))+-123.79905651572749\*ln(1+T5(x30))+224.32895982520978\*ln(1+T6(x30))+-116.53386189013663\*ln(1+T7(x30)))-1)+1.000674926090121\*ln(1+exp(+29.500854346985555\*ln(1+T0(x31))+28.064631829361407\*ln(1+T1(x31))+26.628417242848098\*ln(1+T2(x31))+20.180255851073184\*ln(1+T3(x31))+4.616164750587913\*ln(1+T4(x31))+-15.0322697915982\*ln(1+T5(x31))+-34.32185266920681\*ln(1+T6(x31))+-52.28811215385552\*ln(1+T7(x31)))-1)+1.000518984536183\*ln(1+exp(+29.500854346985555\*ln(1+T0(x32))+-4.541621070374992\*ln(1+T1(x32))+-38.58409804044643\*ln(1+T2(x32))+108.7404978148402\*ln(1+T3(x32))+-122.15195798118881\*ln(1+T4(x32))+-74.46412706531274\*ln(1+T5(x32))+312.91779054234513\*ln(1+T6(x32))+-181.10355922606834\*ln(1+T7(x32)))-1))-1

Ф21=+exp(+1.0000176149128848\*ln(1+exp(+-2.181882861709858\*ln(1+T0(x10))+4.92115361201848\*ln(1+T1(x10))+12.024190085850053\*ln(1+T2(x10))+-11.557578947803304\*ln(1+T3(x10))+-7.305872068785254\*ln(1+T4(x10))+15.160109609230659\*ln(1+T5(x10))+-8.59664546795218\*ln(1+T6(x10))+1.4389062818158873\*ln(1+T7(x10)))-1)+1.0000197236726038\*ln(1+exp(+-2.181882861709858\*ln(1+T0(x11))+-1.9070699593984872\*ln(1+T1(x11))+-1.6322570573191686\*ln(1+T2(x11))+2.301881527374795\*ln(1+T3(x11))+1.0338143336905485\*ln(1+T4(x11))+-4.158625744992928\*ln(1+T5(x11))+4.452541534602822\*ln(1+T6(x11))+-1.737700875187219\*ln(1+T7(x11)))-1))-1

Ф22=+exp(+0.9998171010358119\*ln(1+exp(+7.777468214075455\*ln(1+T0(x20))+-36.44726602726387\*ln(1+T1(x20))+-80.6720002858276\*ln(1+T2(x20))+128.17178818167355\*ln(1+T3(x20))+-82.86046870624475\*ln(1+T4(x20))+49.24027092495042\*ln(1+T5(x20))+-23.46103739549334\*ln(1+T6(x20))+5.966520968914767\*ln(1+T7(x20)))-1)+0.9998188534555269\*ln(1+exp(+7.777468214075455\*ln(1+T0(x21))+25.846082234968463\*ln(1+T1(x21))+43.914699170018494\*ln(1+T2(x21))+-79.88061256335635\*ln(1+T3(x21))+59.74531652165636\*ln(1+T4(x21))+-35.39178807490148\*ln(1+T5(x21))+11.568416873589674\*ln(1+T6(x21))+-0.8091645263069945\*ln(1+T7(x21)))-1))-1

Ф23=+exp(+0.9964349260169533\*ln(1+exp(+143.85562255220825\*ln(1+T0(x30))+-81.77097707577096\*ln(1+T1(x30))+-307.2501393782645\*ln(1+T2(x30))+512.1522046382631\*ln(1+T3(x30))+-712.9534804378683\*ln(1+T4(x30))+1289.6282237186126\*ln(1+T5(x30))+-1644.5275356238267\*ln(1+T6(x30))+801.1708376692931\*ln(1+T7(x30)))-1)+0.9964332900044168\*ln(1+exp(+143.85562255220825\*ln(1+T0(x31))+67.70231821645648\*ln(1+T1(x31))+-8.394617444173624\*ln(1+T2(x31))+-54.01362159230008\*ln(1+T3(x31))+-44.46772853902575\*ln(1+T4(x31))+-10.095282749714658\*ln(1+T5(x31))+22.056251835539836\*ln(1+T6(x31))+46.017589157079776\*ln(1+T7(x31)))-1)+0.9964306684979773\*ln(1+exp(+143.85562255220825\*ln(1+T0(x32))+-369.41463853565114\*ln(1+T1(x32))+-882.5320054352666\*ln(1+T2(x32))+1569.5154740031605\*ln(1+T3(x32))+-1296.379151854587\*ln(1+T4(x32))+537.6340053510274\*ln(1+T5(x32))+69.92363493206844\*ln(1+T6(x32))+-70.22126890106111\*ln(1+T7(x32)))-1))-1

Ф31=+exp(+1.0020143075506849\*ln(1+exp(+-2.1733962322649654\*ln(1+T0(x10))+4.902191700861223\*ln(1+T1(x10))+11.977779634069579\*ln(1+T2(x10))+-11.83497902398336\*ln(1+T3(x10))+-6.912840937191477\*ln(1+T4(x10))+17.776334102863355\*ln(1+T5(x10))+-14.660493813597842\*ln(1+T6(x10))+4.757549766771358\*ln(1+T7(x10)))-1)+1.002117227713355\*ln(1+exp(+-2.1733962322649654\*ln(1+T0(x11))+-1.8454198829663737\*ln(1+T1(x11))+-1.517443533750061\*ln(1+T2(x11))+2.0106817942693427\*ln(1+T3(x11))+1.1657677146900756\*ln(1+T4(x11))+-3.590912055801638\*ln(1+T5(x11))+3.2578788355532278\*ln(1+T6(x11))+-1.1069281586383286\*ln(1+T7(x11)))-1))-1

Ф32=+exp(+1.000835011947189\*ln(1+exp(+0.8071305258059297\*ln(1+T0(x20))+-0.017017907042396228\*ln(1+T1(x20))+-0.8411663398899353\*ln(1+T2(x20))+0.17872238843278174\*ln(1+T3(x20))+1.0638737509039984\*ln(1+T4(x20))+-0.6460131101194501\*ln(1+T5(x20))+-0.5343571788922898\*ln(1+T6(x20))+0.680811251176263\*ln(1+T7(x20)))-1)+1.0038104205221057\*ln(1+exp(+0.8071305258059297\*ln(1+T0(x21))+-0.6234702951320609\*ln(1+T1(x21))+-2.0540711160739256\*ln(1+T2(x21))+0.0931162263847704\*ln(1+T3(x21))+3.7692677658267666\*ln(1+T4(x21))+-0.75359343398973\*ln(1+T5(x21))+-4.464557414773323\*ln(1+T6(x21))+2.8455267264680484\*ln(1+T7(x21)))-1))-1

Ф33=+exp(+1.000001671735292\*ln(1+exp(+177.9605073711452\*ln(1+T0(x30))+-312.1281429691977\*ln(1+T1(x30))+-802.2148353194368\*ln(1+T2(x30))+1439.5662339374583\*ln(1+T3(x30))+-1553.930084848189\*ln(1+T4(x30))+1941.1101951985374\*ln(1+T5(x30))+-1850.513748167983\*ln(1+T6(x30))+764.566540619912\*ln(1+T7(x30)))-1)+1.0000016592240633\*ln(1+exp(+177.9605073711452\*ln(1+T0(x31))+101.51656809814367\*ln(1+T1(x31))+25.08428692033226\*ln(1+T2(x31))+-27.115859990943296\*ln(1+T3(x31))+-35.28622622387218\*ln(1+T4(x31))+-23.743626539758342\*ln(1+T5(x31))+-13.916614201306235\*ln(1+T6(x31))+-10.498974098454187\*ln(1+T7(x31)))-1)+1.0000017047948546\*ln(1+exp(+177.9605073711452\*ln(1+T0(x32))+-194.25740830485233\*ln(1+T1(x32))+-566.4725605085316\*ln(1+T2(x32))+975.7116893339941\*ln(1+T3(x32))+-785.6278408197325\*ln(1+T4(x32))+256.2145117873805\*ln(1+T5(x32))+146.48769364706817\*ln(1+T6(x32))+-88.0622411749155\*ln(1+T7(x32)))-1))-1

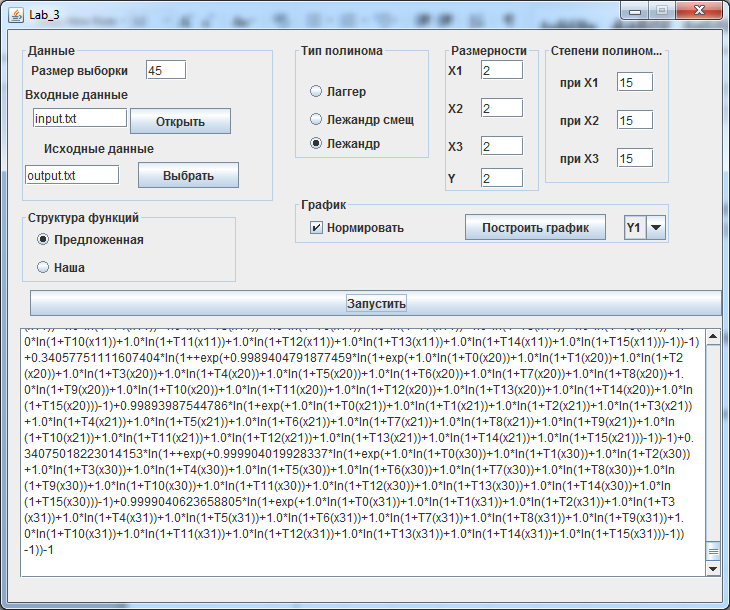
Ф0=exp(+0.1107413509360698\*ln(1++exp(+1.0002347156990836\*ln(1+exp(+3.838110893199928\*ln(1+T0(x10))+1.5309520249489794\*ln(1+T1(x10))+-0.7762068335261156\*ln(1+T2(x10))+0.6850741510062301\*ln(1+T3(x10))+-2.279384942782013\*ln(1+T4(x10))+4.242414283199751\*ln(1+T5(x10))+-4.292187363020475\*ln(1+T6(x10))+1.6997907701386648\*ln(1+T7(x10)))-1)+1.0002450026563976\*ln(1+exp(+3.838110893199928\*ln(1+T0(x11))+-6.3722266722308865\*ln(1+T1(x11))+-16.58256423172599\*ln(1+T2(x11))+18.29409835573527\*ln(1+T3(x11))+7.876481181841437\*ln(1+T4(x11))+-33.617619247378165\*ln(1+T5(x11))+35.8793044856896\*ln(1+T6(x11))+-14.0264752503876\*ln(1+T7(x11)))-1))-1)+-0.023365687507388315\*ln(1++exp(+0.9999949176853706\*ln(1+exp(+45.58847865637187\*ln(1+T0(x20))+-50.617665821879115\*ln(1+T1(x20))+-146.82381026937128\*ln(1+T2(x20))+229.29248925498138\*ln(1+T3(x20))+-158.2215202518411\*ln(1+T4(x20))+100.97444892226179\*ln(1+T5(x20))+-51.22397100803834\*ln(1+T6(x20))+13.106055057733354\*ln(1+T7(x20)))-1)+0.9999963471330582\*ln(1+exp(+45.58847865637187\*ln(1+T0(x21))+-17.115340815417426\*ln(1+T1(x21))+-79.81916013139465\*ln(1+T2(x21))+120.30667148691144\*ln(1+T3(x21))+-84.60931969098375\*ln(1+T4(x21))+48.62148927926165\*ln(1+T5(x21))+-16.28706858417513\*ln(1+T6(x21))+1.125060524460816\*ln(1+T7(x21)))-1))-1)+0.914474406115407\*ln(1++exp(+1.0052204351945253\*ln(1+exp(+378.0865952855692\*ln(1+T0(x30))+-17.371246467452302\*ln(1+T1(x30))+-412.3468065046327\*ln(1+T2(x30))+652.6643152540636\*ln(1+T3(x30))+-874.8162833229435\*ln(1+T4(x30))+1411.874938026344\*ln(1+T5(x30))+-1612.5545996701737\*ln(1+T6(x30))+737.4285877771393\*ln(1+T7(x30)))-1)+1.0052311769464735\*ln(1+exp(+378.0865952855692\*ln(1+T0(x31))+206.47620531632762\*ln(1+T1(x31))+35.11069644403024\*ln(1+T2(x31))+-75.77069683875955\*ln(1+T3(x31))+-77.32488086534003\*ln(1+T4(x31))+-29.49328702631603\*ln(1+T5(x31))+14.254514179546605\*ln(1+T6(x31))+41.83010199893179\*ln(1+T7(x31)))-1)+1.0052273673030179\*ln(1+exp(+378.0865952855692\*ln(1+T0(x32))+-1119.156263363238\*ln(1+T1(x32))+-2616.083116529668\*ln(1+T2(x32))+4976.854025396079\*ln(1+T3(x32))+-4977.018458694015\*ln(1+T4(x32))+4087.6303613512637\*ln(1+T5(x32))+-1978.4105977252473\*ln(1+T6(x32))+394.5335156222737\*ln(1+T7(x32)))-1))-1))-1

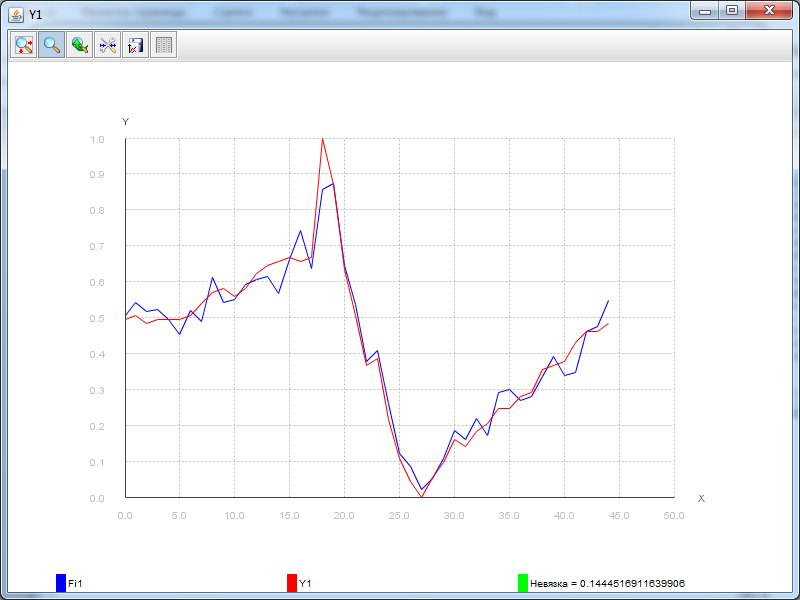
Ф1=exp(+0.18577375598937354\*ln(1++exp(+0.9869145648552234\*ln(1+exp(+18.503035665678645\*ln(1+T0(x10))+-44.949936648926695\*ln(1+T1(x10))+-108.4029090923041\*ln(1+T2(x10))+181.24950965880515\*ln(1+T3(x10))+-137.33439102872822\*ln(1+T4(x10))+98.01502477631169\*ln(1+T5(x10))+-55.04210196521577\*ln(1+T6(x10))+15.795795596990004\*ln(1+T7(x10)))-1)+0.987075190600403\*ln(1+exp(+18.503035665678645\*ln(1+T0(x11))+16.763806540738184\*ln(1+T1(x11))+15.024577330963218\*ln(1+T2(x11))+-24.35522022084423\*ln(1+T3(x11))+0.17902989842808203\*ln(1+T4(x11))+22.795743462926524\*ln(1+T5(x11))+-27.45624808903215\*ln(1+T6(x11))+11.424741650692706\*ln(1+T7(x11)))-1))-1)+0.002667451112625993\*ln(1++exp(+0.7740391863208671\*ln(1+exp(+0.09732534744617177\*ln(1+T0(x20))+0.0755888987815842\*ln(1+T1(x20))+0.05385245011699665\*ln(1+T2(x20))+0.03380192749854132\*ln(1+T3(x20))+0.017841308312352134\*ln(1+T4(x20))+0.006357033086470353\*ln(1+T5(x20))+-0.0019450927975522316\*ln(1+T6(x20))+-0.00803742238432353\*ln(1+T7(x20)))-1)+1.1566698616221043\*ln(1+exp(+0.09732534744617177\*ln(1+T0(x21))+0.07104173640455518\*ln(1+T1(x21))+0.04475812536293872\*ln(1+T2(x21))+0.017745361446817154\*ln(1+T3(x21))+-0.009770345874704147\*ln(1+T4(x21))+-0.03374017343582745\*ln(1+T5(x21))+-0.05118259391097622\*ln(1+T6(x21))+-0.06197732236587362\*ln(1+T7(x21)))-1))-1)+0.8273942850356686\*ln(1++exp(+1.000557858512941\*ln(1+exp(+29.500854346985555\*ln(1+T0(x30))+-46.16824015385558\*ln(1+T1(x30))+-121.83733745999889\*ln(1+T2(x30))+176.18779905570915\*ln(1+T3(x30))+-47.58401583657103\*ln(1+T4(x30))+-123.79905651572749\*ln(1+T5(x30))+224.32895982520978\*ln(1+T6(x30))+-116.53386189013663\*ln(1+T7(x30)))-1)+1.000674926090121\*ln(1+exp(+29.500854346985555\*ln(1+T0(x31))+28.064631829361407\*ln(1+T1(x31))+26.628417242848098\*ln(1+T2(x31))+20.180255851073184\*ln(1+T3(x31))+4.616164750587913\*ln(1+T4(x31))+-15.0322697915982\*ln(1+T5(x31))+-34.32185266920681\*ln(1+T6(x31))+-52.28811215385552\*ln(1+T7(x31)))-1)+1.000518984536183\*ln(1+exp(+29.500854346985555\*ln(1+T0(x32))+-4.541621070374992\*ln(1+T1(x32))+-38.58409804044643\*ln(1+T2(x32))+108.7404978148402\*ln(1+T3(x32))+-122.15195798118881\*ln(1+T4(x32))+-74.46412706531274\*ln(1+T5(x32))+312.91779054234513\*ln(1+T6(x32))+-181.10355922606834\*ln(1+T7(x32)))-1))-1))-1

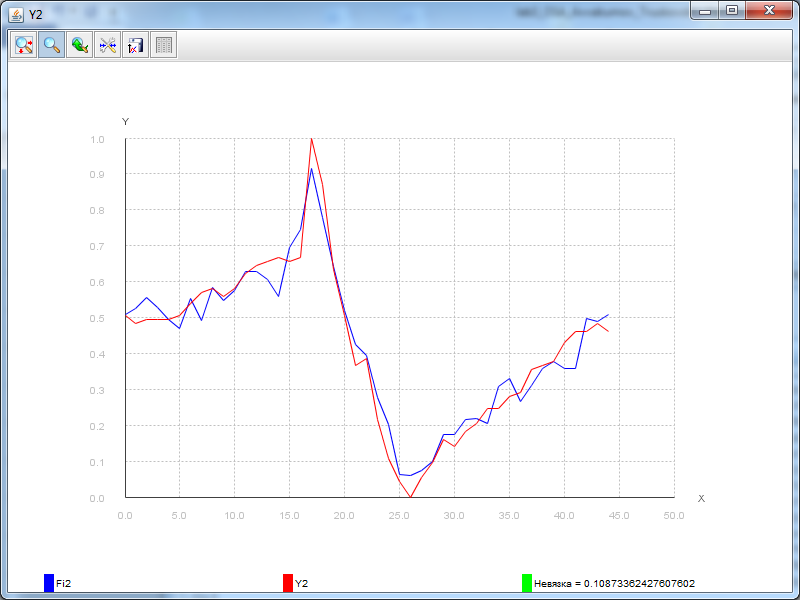
Ф2=exp(+0.20417260374910928\*ln(1++exp(+1.0000176149128848\*ln(1+exp(+-2.181882861709858\*ln(1+T0(x10))+4.92115361201848\*ln(1+T1(x10))+12.024190085850053\*ln(1+T2(x10))+-11.557578947803304\*ln(1+T3(x10))+-7.305872068785254\*ln(1+T4(x10))+15.160109609230659\*ln(1+T5(x10))+-8.59664546795218\*ln(1+T6(x10))+1.4389062818158873\*ln(1+T7(x10)))-1)+1.0000197236726038\*ln(1+exp(+-2.181882861709858\*ln(1+T0(x11))+-1.9070699593984872\*ln(1+T1(x11))+-1.6322570573191686\*ln(1+T2(x11))+2.301881527374795\*ln(1+T3(x11))+1.0338143336905485\*ln(1+T4(x11))+-4.158625744992928\*ln(1+T5(x11))+4.452541534602822\*ln(1+T6(x11))+-1.737700875187219\*ln(1+T7(x11)))-1))-1)+0.02185683751519796\*ln(1++exp(+0.9998171010358119\*ln(1+exp(+7.777468214075455\*ln(1+T0(x20))+-36.44726602726387\*ln(1+T1(x20))+-80.6720002858276\*ln(1+T2(x20))+128.17178818167355\*ln(1+T3(x20))+-82.86046870624475\*ln(1+T4(x20))+49.24027092495042\*ln(1+T5(x20))+-23.46103739549334\*ln(1+T6(x20))+5.966520968914767\*ln(1+T7(x20)))-1)+0.9998188534555269\*ln(1+exp(+7.777468214075455\*ln(1+T0(x21))+25.846082234968463\*ln(1+T1(x21))+43.914699170018494\*ln(1+T2(x21))+-79.88061256335635\*ln(1+T3(x21))+59.74531652165636\*ln(1+T4(x21))+-35.39178807490148\*ln(1+T5(x21))+11.568416873589674\*ln(1+T6(x21))+-0.8091645263069945\*ln(1+T7(x21)))-1))-1)+0.8057939706252961\*ln(1++exp(+0.9964349260169533\*ln(1+exp(+143.85562255220825\*ln(1+T0(x30))+-81.77097707577096\*ln(1+T1(x30))+-307.2501393782645\*ln(1+T2(x30))+512.1522046382631\*ln(1+T3(x30))+-712.9534804378683\*ln(1+T4(x30))+1289.6282237186126\*ln(1+T5(x30))+-1644.5275356238267\*ln(1+T6(x30))+801.1708376692931\*ln(1+T7(x30)))-1)+0.9964332900044168\*ln(1+exp(+143.85562255220825\*ln(1+T0(x31))+67.70231821645648\*ln(1+T1(x31))+-8.394617444173624\*ln(1+T2(x31))+-54.01362159230008\*ln(1+T3(x31))+-44.46772853902575\*ln(1+T4(x31))+-10.095282749714658\*ln(1+T5(x31))+22.056251835539836\*ln(1+T6(x31))+46.017589157079776\*ln(1+T7(x31)))-1)+0.9964306684979773\*ln(1+exp(+143.85562255220825\*ln(1+T0(x32))+-369.41463853565114\*ln(1+T1(x32))+-882.5320054352666\*ln(1+T2(x32))+1569.5154740031605\*ln(1+T3(x32))+-1296.379151854587\*ln(1+T4(x32))+537.6340053510274\*ln(1+T5(x32))+69.92363493206844\*ln(1+T6(x32))+-70.22126890106111\*ln(1+T7(x32)))-1))-1))-1

Ф3=exp(+0.27177924640691004\*ln(1++exp(+1.0020143075506849\*ln(1+exp(+-2.1733962322649654\*ln(1+T0(x10))+4.902191700861223\*ln(1+T1(x10))+11.977779634069579\*ln(1+T2(x10))+-11.83497902398336\*ln(1+T3(x10))+-6.912840937191477\*ln(1+T4(x10))+17.776334102863355\*ln(1+T5(x10))+-14.660493813597842\*ln(1+T6(x10))+4.757549766771358\*ln(1+T7(x10)))-1)+1.002117227713355\*ln(1+exp(+-2.1733962322649654\*ln(1+T0(x11))+-1.8454198829663737\*ln(1+T1(x11))+-1.517443533750061\*ln(1+T2(x11))+2.0106817942693427\*ln(1+T3(x11))+1.1657677146900756\*ln(1+T4(x11))+-3.590912055801638\*ln(1+T5(x11))+3.2578788355532278\*ln(1+T6(x11))+-1.1069281586383286\*ln(1+T7(x11)))-1))-1)+0.02823426464721079\*ln(1++exp(+1.000835011947189\*ln(1+exp(+0.8071305258059297\*ln(1+T0(x20))+-0.017017907042396228\*ln(1+T1(x20))+-0.8411663398899353\*ln(1+T2(x20))+0.17872238843278174\*ln(1+T3(x20))+1.0638737509039984\*ln(1+T4(x20))+-0.6460131101194501\*ln(1+T5(x20))+-0.5343571788922898\*ln(1+T6(x20))+0.680811251176263\*ln(1+T7(x20)))-1)+1.0038104205221057\*ln(1+exp(+0.8071305258059297\*ln(1+T0(x21))+-0.6234702951320609\*ln(1+T1(x21))+-2.0540711160739256\*ln(1+T2(x21))+0.0931162263847704\*ln(1+T3(x21))+3.7692677658267666\*ln(1+T4(x21))+-0.75359343398973\*ln(1+T5(x21))+-4.464557414773323\*ln(1+T6(x21))+2.8455267264680484\*ln(1+T7(x21)))-1))-1)+0.7243733288970752\*ln(1++exp(+1.000001671735292\*ln(1+exp(+177.9605073711452\*ln(1+T0(x30))+-312.1281429691977\*ln(1+T1(x30))+-802.2148353194368\*ln(1+T2(x30))+1439.5662339374583\*ln(1+T3(x30))+-1553.930084848189\*ln(1+T4(x30))+1941.1101951985374\*ln(1+T5(x30))+-1850.513748167983\*ln(1+T6(x30))+764.566540619912\*ln(1+T7(x30)))-1)+1.0000016592240633\*ln(1+exp(+177.9605073711452\*ln(1+T0(x31))+101.51656809814367\*ln(1+T1(x31))+25.08428692033226\*ln(1+T2(x31))+-27.115859990943296\*ln(1+T3(x31))+-35.28622622387218\*ln(1+T4(x31))+-23.743626539758342\*ln(1+T5(x31))+-13.916614201306235\*ln(1+T6(x31))+-10.498974098454187\*ln(1+T7(x31)))-1)+1.0000017047948546\*ln(1+exp(+177.9605073711452\*ln(1+T0(x32))+-194.25740830485233\*ln(1+T1(x32))+-566.4725605085316\*ln(1+T2(x32))+975.7116893339941\*ln(1+T3(x32))+-785.6278408197325\*ln(1+T4(x32))+256.2145117873805\*ln(1+T5(x32))+146.48769364706817\*ln(1+T6(x32))+-88.0622411749155\*ln(1+T7(x32)))-1))-1))-1

Для реальної фізичної задачі оцінювання складових сонячних бурь отримали такі результати:







psi010=exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1

psi011=exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1

psi020=exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1

psi021=exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21))+1.0\*ln(1+T8(x21))+1.0\*ln(1+T9(x21))+1.0\*ln(1+T10(x21))+1.0\*ln(1+T11(x21))+1.0\*ln(1+T12(x21))+1.0\*ln(1+T13(x21))+1.0\*ln(1+T14(x21))+1.0\*ln(1+T15(x21)))-1

psi030=exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1

psi031=exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1

psi110=exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1

psi111=exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1

psi120=exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1

psi121=exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21))+1.0\*ln(1+T8(x21))+1.0\*ln(1+T9(x21))+1.0\*ln(1+T10(x21))+1.0\*ln(1+T11(x21))+1.0\*ln(1+T12(x21))+1.0\*ln(1+T13(x21))+1.0\*ln(1+T14(x21))+1.0\*ln(1+T15(x21)))-1

psi130=exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1

psi131=exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1

Ф01=+exp(+0.9910160081256251\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1)+0.9910185983879853\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1))-1

Ф02=+exp(+1.0021644846630842\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1)+1.002163975135881\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21))+1.0\*ln(1+T8(x21))+1.0\*ln(1+T9(x21))+1.0\*ln(1+T10(x21))+1.0\*ln(1+T11(x21))+1.0\*ln(1+T12(x21))+1.0\*ln(1+T13(x21))+1.0\*ln(1+T14(x21))+1.0\*ln(1+T15(x21)))-1))-1

Ф03=+exp(+0.9999974380390728\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1)+0.9999974392460361\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1))-1

Ф11=+exp(+0.9981945907928329\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1)+0.9981948024310728\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1))-1

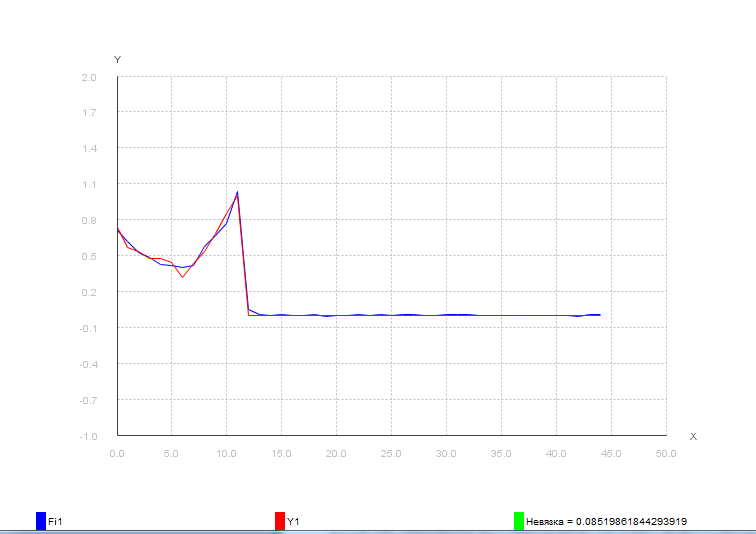
Ф12=+exp(+0.9989404791877459\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1)+0.99893987544786\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21))+1.0\*ln(1+T8(x21))+1.0\*ln(1+T9(x21))+1.0\*ln(1+T10(x21))+1.0\*ln(1+T11(x21))+1.0\*ln(1+T12(x21))+1.0\*ln(1+T13(x21))+1.0\*ln(1+T14(x21))+1.0\*ln(1+T15(x21)))-1))-1

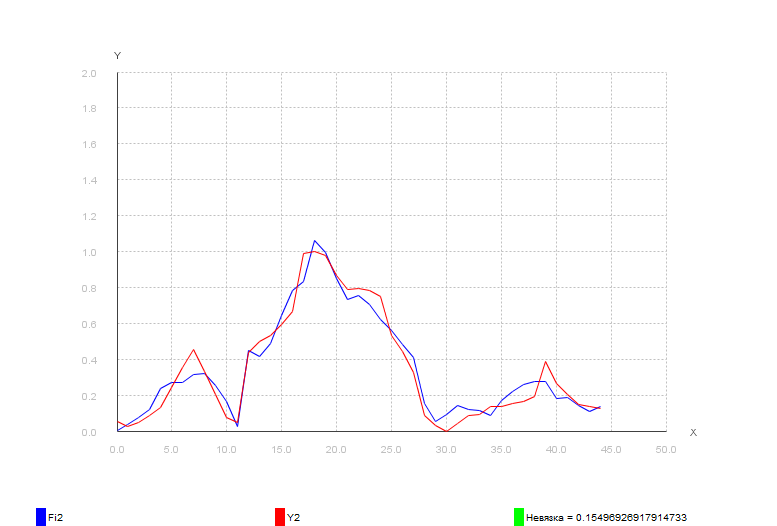
Ф13=+exp(+0.999904019928337\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1)+0.9999040623658805\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1))-1

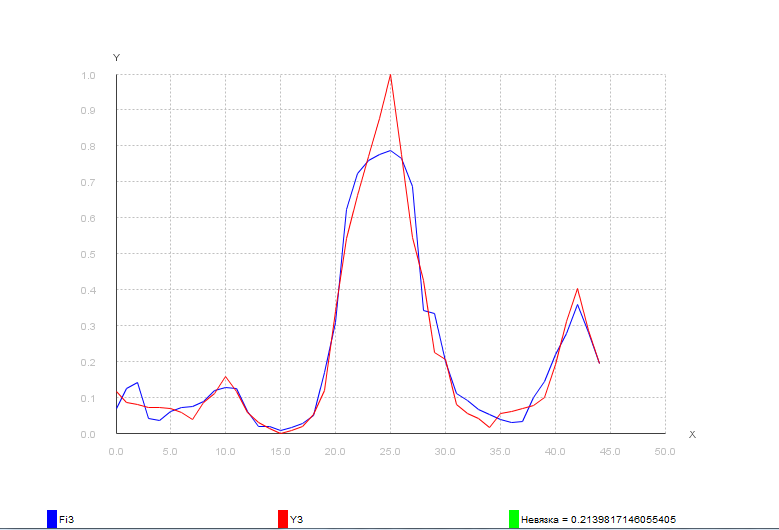
Ф0=exp(+0.33304865493322056\*ln(1++exp(+0.9910160081256251\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1)+0.9910185983879853\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1))-1)+0.3366917748565443\*ln(1++exp(+1.0021644846630842\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1)+1.002163975135881\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21))+1.0\*ln(1+T8(x21))+1.0\*ln(1+T9(x21))+1.0\*ln(1+T10(x21))+1.0\*ln(1+T11(x21))+1.0\*ln(1+T12(x21))+1.0\*ln(1+T13(x21))+1.0\*ln(1+T14(x21))+1.0\*ln(1+T15(x21)))-1))-1)+0.33835687144612775\*ln(1++exp(+0.9999974380390728\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1)+0.9999974392460361\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1))-1))-1

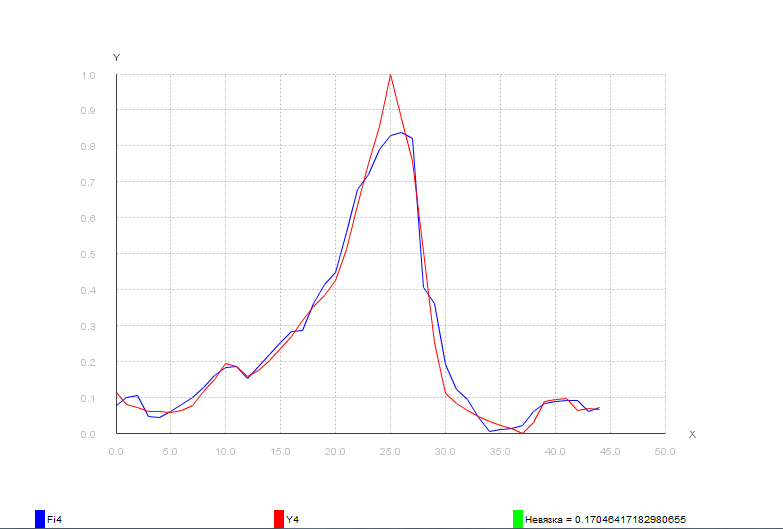
Ф1=exp(+0.33043364108351614\*ln(1++exp(+0.9981945907928329\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1)+0.9981948024310728\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1))-1)+0.34057751111607404\*ln(1++exp(+0.9989404791877459\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1)+0.99893987544786\*ln(1+exp(+1.0\*ln(1+T0(x21))+1.0\*ln(1+T1(x21))+1.0\*ln(1+T2(x21))+1.0\*ln(1+T3(x21))+1.0\*ln(1+T4(x21))+1.0\*ln(1+T5(x21))+1.0\*ln(1+T6(x21))+1.0\*ln(1+T7(x21))+1.0\*ln(1+T8(x21))+1.0\*ln(1+T9(x21))+1.0\*ln(1+T10(x21))+1.0\*ln(1+T11(x21))+1.0\*ln(1+T12(x21))+1.0\*ln(1+T13(x21))+1.0\*ln(1+T14(x21))+1.0\*ln(1+T15(x21)))-1))-1)+0.34075018223014153\*ln(1++exp(+0.999904019928337\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1)+0.9999040623658805\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1))-1))-1

**2.** *Построить прогнозные значения*

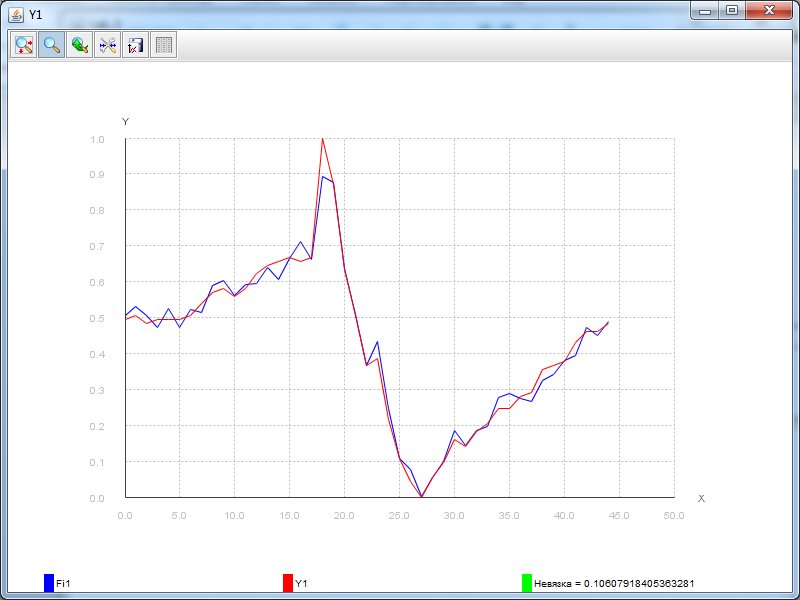


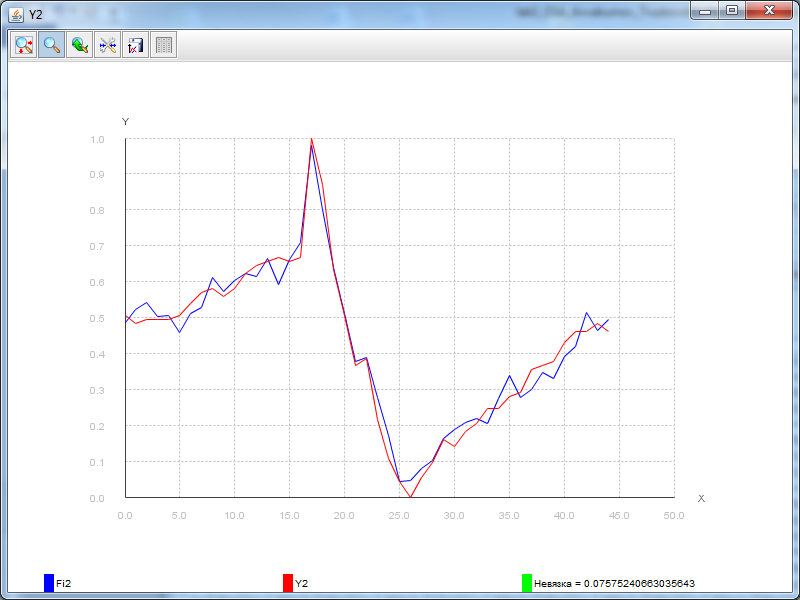






Для задачі оцінювання складових сонячних бурь:





1. Предложить свой вариант дискретной выборки и  (для реальной задачи) и построить в мультипликативной форме приближающие функции , .

Взявши вибірку, яка показує залежність потужності обчислювальної машини залежно від набору параметрів(розглядалась у роботі № 2), отримали такі результати:

psi010=exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1

psi011=exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1

psi012=exp(+1.0\*ln(1+T0(x12))+1.0\*ln(1+T1(x12))+1.0\*ln(1+T2(x12))+1.0\*ln(1+T3(x12))+1.0\*ln(1+T4(x12))+1.0\*ln(1+T5(x12))+1.0\*ln(1+T6(x12))+1.0\*ln(1+T7(x12))+1.0\*ln(1+T8(x12))+1.0\*ln(1+T9(x12))+1.0\*ln(1+T10(x12))+1.0\*ln(1+T11(x12))+1.0\*ln(1+T12(x12))+1.0\*ln(1+T13(x12))+1.0\*ln(1+T14(x12))+1.0\*ln(1+T15(x12)))-1

psi020=exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1

psi030=exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1

psi031=exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1

psi110=exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1

psi111=exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1

psi112=exp(+1.0\*ln(1+T0(x12))+1.0\*ln(1+T1(x12))+1.0\*ln(1+T2(x12))+1.0\*ln(1+T3(x12))+1.0\*ln(1+T4(x12))+1.0\*ln(1+T5(x12))+1.0\*ln(1+T6(x12))+1.0\*ln(1+T7(x12))+1.0\*ln(1+T8(x12))+1.0\*ln(1+T9(x12))+1.0\*ln(1+T10(x12))+1.0\*ln(1+T11(x12))+1.0\*ln(1+T12(x12))+1.0\*ln(1+T13(x12))+1.0\*ln(1+T14(x12))+1.0\*ln(1+T15(x12)))-1

psi120=exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1

psi130=exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1

psi131=exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1

Ф01=+exp(+0.9999971315671543\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1)+0.9999972521778633\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1)+0.9999970919085651\*ln(1+exp(+1.0\*ln(1+T0(x12))+1.0\*ln(1+T1(x12))+1.0\*ln(1+T2(x12))+1.0\*ln(1+T3(x12))+1.0\*ln(1+T4(x12))+1.0\*ln(1+T5(x12))+1.0\*ln(1+T6(x12))+1.0\*ln(1+T7(x12))+1.0\*ln(1+T8(x12))+1.0\*ln(1+T9(x12))+1.0\*ln(1+T10(x12))+1.0\*ln(1+T11(x12))+1.0\*ln(1+T12(x12))+1.0\*ln(1+T13(x12))+1.0\*ln(1+T14(x12))+1.0\*ln(1+T15(x12)))-1))-1

Ф02=+exp(+0.9999928779507968\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1))-1

Ф03=+exp(+1.007599154416144\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1)+1.0075991582847585\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1))-1

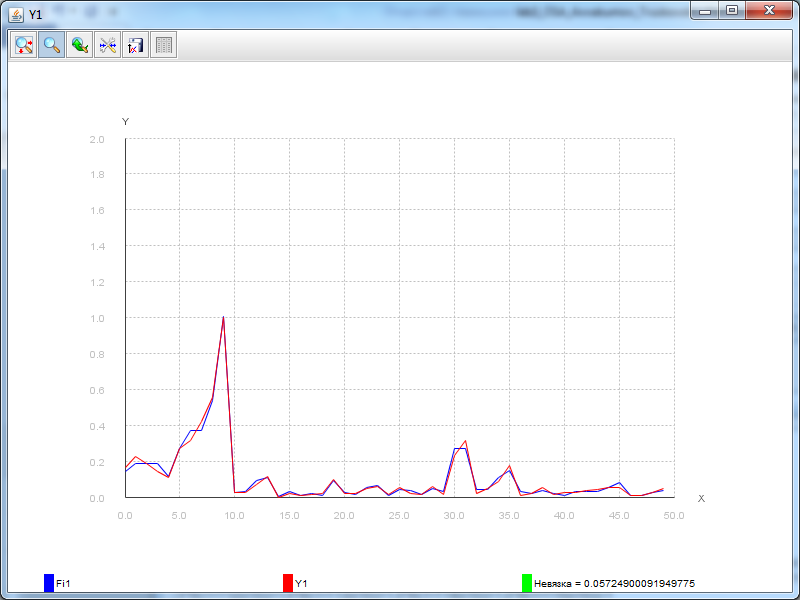
Ф11=+exp(+1.0005524226192928\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1)+0.9993953262645365\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1)+1.0004658594584235\*ln(1+exp(+1.0\*ln(1+T0(x12))+1.0\*ln(1+T1(x12))+1.0\*ln(1+T2(x12))+1.0\*ln(1+T3(x12))+1.0\*ln(1+T4(x12))+1.0\*ln(1+T5(x12))+1.0\*ln(1+T6(x12))+1.0\*ln(1+T7(x12))+1.0\*ln(1+T8(x12))+1.0\*ln(1+T9(x12))+1.0\*ln(1+T10(x12))+1.0\*ln(1+T11(x12))+1.0\*ln(1+T12(x12))+1.0\*ln(1+T13(x12))+1.0\*ln(1+T14(x12))+1.0\*ln(1+T15(x12)))-1))-1

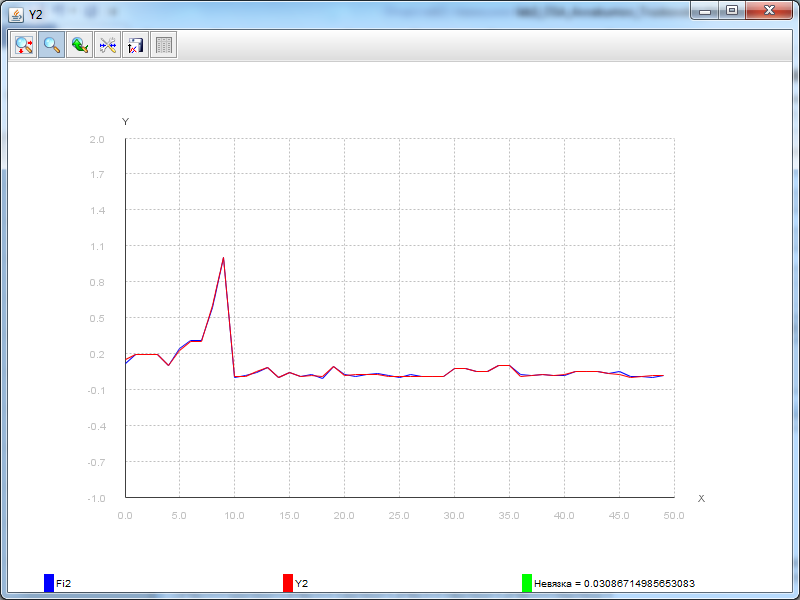
Ф12=+exp(+0.9998814961474154\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1))-1

Ф13=+exp(+0.980043902973136\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1)+0.9907666566072366\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1))-1

Ф0=exp(+0.9136813866879031\*ln(1++exp(+0.9999971315671543\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1)+0.9999972521778633\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1)+0.9999970919085651\*ln(1+exp(+1.0\*ln(1+T0(x12))+1.0\*ln(1+T1(x12))+1.0\*ln(1+T2(x12))+1.0\*ln(1+T3(x12))+1.0\*ln(1+T4(x12))+1.0\*ln(1+T5(x12))+1.0\*ln(1+T6(x12))+1.0\*ln(1+T7(x12))+1.0\*ln(1+T8(x12))+1.0\*ln(1+T9(x12))+1.0\*ln(1+T10(x12))+1.0\*ln(1+T11(x12))+1.0\*ln(1+T12(x12))+1.0\*ln(1+T13(x12))+1.0\*ln(1+T14(x12))+1.0\*ln(1+T15(x12)))-1))-1)+0.055288706863208914\*ln(1++exp(+0.9999928779507968\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1))-1)+0.03692596015454823\*ln(1++exp(+1.007599154416144\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1)+1.0075991582847585\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1))-1))-1

Ф1=exp(+0.9402210014390221\*ln(1++exp(+1.0005524226192928\*ln(1+exp(+1.0\*ln(1+T0(x10))+1.0\*ln(1+T1(x10))+1.0\*ln(1+T2(x10))+1.0\*ln(1+T3(x10))+1.0\*ln(1+T4(x10))+1.0\*ln(1+T5(x10))+1.0\*ln(1+T6(x10))+1.0\*ln(1+T7(x10))+1.0\*ln(1+T8(x10))+1.0\*ln(1+T9(x10))+1.0\*ln(1+T10(x10))+1.0\*ln(1+T11(x10))+1.0\*ln(1+T12(x10))+1.0\*ln(1+T13(x10))+1.0\*ln(1+T14(x10))+1.0\*ln(1+T15(x10)))-1)+0.9993953262645365\*ln(1+exp(+1.0\*ln(1+T0(x11))+1.0\*ln(1+T1(x11))+1.0\*ln(1+T2(x11))+1.0\*ln(1+T3(x11))+1.0\*ln(1+T4(x11))+1.0\*ln(1+T5(x11))+1.0\*ln(1+T6(x11))+1.0\*ln(1+T7(x11))+1.0\*ln(1+T8(x11))+1.0\*ln(1+T9(x11))+1.0\*ln(1+T10(x11))+1.0\*ln(1+T11(x11))+1.0\*ln(1+T12(x11))+1.0\*ln(1+T13(x11))+1.0\*ln(1+T14(x11))+1.0\*ln(1+T15(x11)))-1)+1.0004658594584235\*ln(1+exp(+1.0\*ln(1+T0(x12))+1.0\*ln(1+T1(x12))+1.0\*ln(1+T2(x12))+1.0\*ln(1+T3(x12))+1.0\*ln(1+T4(x12))+1.0\*ln(1+T5(x12))+1.0\*ln(1+T6(x12))+1.0\*ln(1+T7(x12))+1.0\*ln(1+T8(x12))+1.0\*ln(1+T9(x12))+1.0\*ln(1+T10(x12))+1.0\*ln(1+T11(x12))+1.0\*ln(1+T12(x12))+1.0\*ln(1+T13(x12))+1.0\*ln(1+T14(x12))+1.0\*ln(1+T15(x12)))-1))-1)+0.03971525530599987\*ln(1++exp(+0.9998814961474154\*ln(1+exp(+1.0\*ln(1+T0(x20))+1.0\*ln(1+T1(x20))+1.0\*ln(1+T2(x20))+1.0\*ln(1+T3(x20))+1.0\*ln(1+T4(x20))+1.0\*ln(1+T5(x20))+1.0\*ln(1+T6(x20))+1.0\*ln(1+T7(x20))+1.0\*ln(1+T8(x20))+1.0\*ln(1+T9(x20))+1.0\*ln(1+T10(x20))+1.0\*ln(1+T11(x20))+1.0\*ln(1+T12(x20))+1.0\*ln(1+T13(x20))+1.0\*ln(1+T14(x20))+1.0\*ln(1+T15(x20)))-1))-1)+0.023240330547478604\*ln(1++exp(+0.980043902973136\*ln(1+exp(+1.0\*ln(1+T0(x30))+1.0\*ln(1+T1(x30))+1.0\*ln(1+T2(x30))+1.0\*ln(1+T3(x30))+1.0\*ln(1+T4(x30))+1.0\*ln(1+T5(x30))+1.0\*ln(1+T6(x30))+1.0\*ln(1+T7(x30))+1.0\*ln(1+T8(x30))+1.0\*ln(1+T9(x30))+1.0\*ln(1+T10(x30))+1.0\*ln(1+T11(x30))+1.0\*ln(1+T12(x30))+1.0\*ln(1+T13(x30))+1.0\*ln(1+T14(x30))+1.0\*ln(1+T15(x30)))-1)+0.9907666566072366\*ln(1+exp(+1.0\*ln(1+T0(x31))+1.0\*ln(1+T1(x31))+1.0\*ln(1+T2(x31))+1.0\*ln(1+T3(x31))+1.0\*ln(1+T4(x31))+1.0\*ln(1+T5(x31))+1.0\*ln(1+T6(x31))+1.0\*ln(1+T7(x31))+1.0\*ln(1+T8(x31))+1.0\*ln(1+T9(x31))+1.0\*ln(1+T10(x31))+1.0\*ln(1+T11(x31))+1.0\*ln(1+T12(x31))+1.0\*ln(1+T13(x31))+1.0\*ln(1+T14(x31))+1.0\*ln(1+T15(x31)))-1))-1))-1





1. Предложить свой вариант структуры функций ,  и построить в мультипликативной форме приближающие функции , 

Нами запропонована така структура функцій:

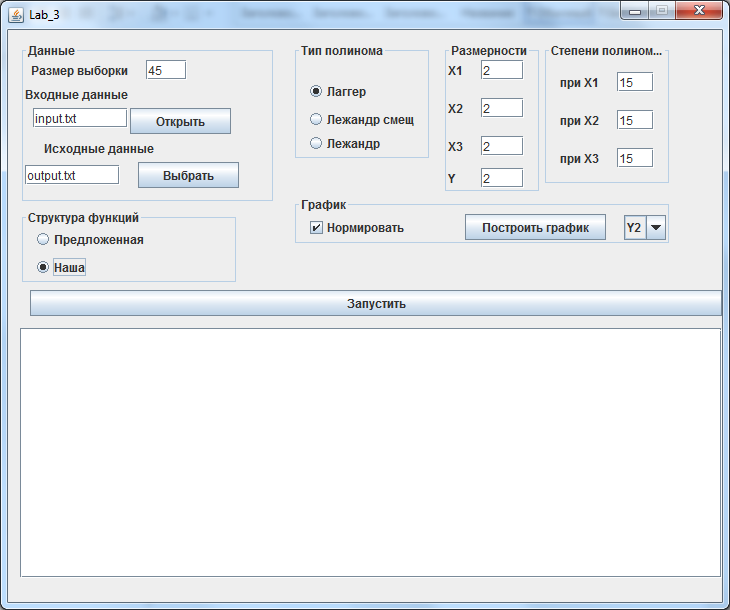
;

;



Цим ми добились зниження помилки обрахунку обчислення на кожному рівні. Це пояснюється тим, що у машинному варіанті представлення чисел мантиса є скінченною, і при піднесенні до степеня накопичується значна похибка. А з використанням функції arcsin(x) ми значно покращили результат, хоча, можливо, існують більш оптимальні варіанти.

Для реальної фізичної задачі оцінювання складових сонячних бурь з використанням запропонованої функції маємо такі результати :

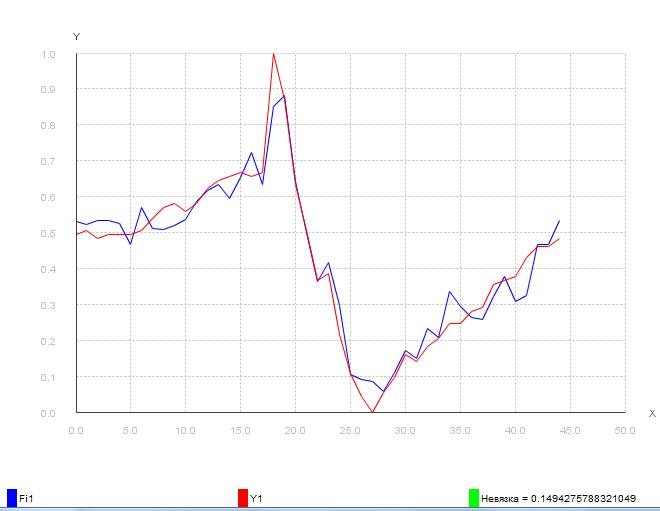


Результати для y1:

З використанням нашої структури функцій :



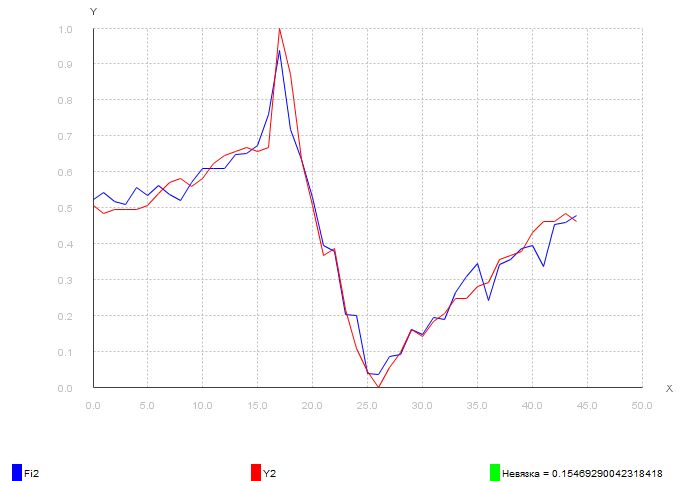
З використанням структури функцій, запропонованої у варіанті:



Результати для y2:

З використанням нашої структури функцій :

З використанням структури функцій, запропонованої у варіанті:



Отже, як бачимо, ми домоглися зменшення нев’язки більше ніж на 0,033 і на 0,038 для у1 та у2 відповідно.

Наближуюча функція в мультиплікативній формі:

psi010=exp(+1.0\*ln(1+arcsin(T0(x10)))+1.0\*ln(1+arcsin(T1(x10)))+1.0\*ln(1+arcsin(T2(x10)))+1.0\*ln(1+arcsin(T3(x10)))+1.0\*ln(1+arcsin(T4(x10)))+1.0\*ln(1+arcsin(T5(x10)))+1.0\*ln(1+arcsin(T6(x10)))+1.0\*ln(1+arcsin(T7(x10)))+1.0\*ln(1+arcsin(T8(x10)))+1.0\*ln(1+arcsin(T9(x10)))+1.0\*ln(1+arcsin(T10(x10)))+1.0\*ln(1+arcsin(T11(x10)))+1.0\*ln(1+arcsin(T12(x10)))+1.0\*ln(1+arcsin(T13(x10)))+1.0\*ln(1+arcsin(T14(x10)))+1.0\*ln(1+arcsin(T15(x10))))-1

psi011=exp(+1.0\*ln(1+arcsin(T0(x11)))+1.0\*ln(1+arcsin(T1(x11)))+1.0\*ln(1+arcsin(T2(x11)))+1.0\*ln(1+arcsin(T3(x11)))+1.0\*ln(1+arcsin(T4(x11)))+1.0\*ln(1+arcsin(T5(x11)))+1.0\*ln(1+arcsin(T6(x11)))+1.0\*ln(1+arcsin(T7(x11)))+1.0\*ln(1+arcsin(T8(x11)))+1.0\*ln(1+arcsin(T9(x11)))+1.0\*ln(1+arcsin(T10(x11)))+1.0\*ln(1+arcsin(T11(x11)))+1.0\*ln(1+arcsin(T12(x11)))+1.0\*ln(1+arcsin(T13(x11)))+1.0\*ln(1+arcsin(T14(x11)))+1.0\*ln(1+arcsin(T15(x11))))-1

psi020=exp(+1.0\*ln(1+arcsin(T0(x20)))+1.0\*ln(1+arcsin(T1(x20)))+1.0\*ln(1+arcsin(T2(x20)))+1.0\*ln(1+arcsin(T3(x20)))+1.0\*ln(1+arcsin(T4(x20)))+1.0\*ln(1+arcsin(T5(x20)))+1.0\*ln(1+arcsin(T6(x20)))+1.0\*ln(1+arcsin(T7(x20)))+1.0\*ln(1+arcsin(T8(x20)))+1.0\*ln(1+arcsin(T9(x20)))+1.0\*ln(1+arcsin(T10(x20)))+1.0\*ln(1+arcsin(T11(x20)))+1.0\*ln(1+arcsin(T12(x20)))+1.0\*ln(1+arcsin(T13(x20)))+1.0\*ln(1+arcsin(T14(x20)))+1.0\*ln(1+arcsin(T15(x20))))-1

psi021=exp(+1.0\*ln(1+arcsin(T0(x21)))+1.0\*ln(1+arcsin(T1(x21)))+1.0\*ln(1+arcsin(T2(x21)))+1.0\*ln(1+arcsin(T3(x21)))+1.0\*ln(1+arcsin(T4(x21)))+1.0\*ln(1+arcsin(T5(x21)))+1.0\*ln(1+arcsin(T6(x21)))+1.0\*ln(1+arcsin(T7(x21)))+1.0\*ln(1+arcsin(T8(x21)))+1.0\*ln(1+arcsin(T9(x21)))+1.0\*ln(1+arcsin(T10(x21)))+1.0\*ln(1+arcsin(T11(x21)))+1.0\*ln(1+arcsin(T12(x21)))+1.0\*ln(1+arcsin(T13(x21)))+1.0\*ln(1+arcsin(T14(x21)))+1.0\*ln(1+arcsin(T15(x21))))-1

psi030=exp(+1.0\*ln(1+arcsin(T0(x30)))+1.0\*ln(1+arcsin(T1(x30)))+1.0\*ln(1+arcsin(T2(x30)))+1.0\*ln(1+arcsin(T3(x30)))+1.0\*ln(1+arcsin(T4(x30)))+1.0\*ln(1+arcsin(T5(x30)))+1.0\*ln(1+arcsin(T6(x30)))+1.0\*ln(1+arcsin(T7(x30)))+1.0\*ln(1+arcsin(T8(x30)))+1.0\*ln(1+arcsin(T9(x30)))+1.0\*ln(1+arcsin(T10(x30)))+1.0\*ln(1+arcsin(T11(x30)))+1.0\*ln(1+arcsin(T12(x30)))+1.0\*ln(1+arcsin(T13(x30)))+1.0\*ln(1+arcsin(T14(x30)))+1.0\*ln(1+arcsin(T15(x30))))-1

psi031=exp(+1.0\*ln(1+arcsin(T0(x31)))+1.0\*ln(1+arcsin(T1(x31)))+1.0\*ln(1+arcsin(T2(x31)))+1.0\*ln(1+arcsin(T3(x31)))+1.0\*ln(1+arcsin(T4(x31)))+1.0\*ln(1+arcsin(T5(x31)))+1.0\*ln(1+arcsin(T6(x31)))+1.0\*ln(1+arcsin(T7(x31)))+1.0\*ln(1+arcsin(T8(x31)))+1.0\*ln(1+arcsin(T9(x31)))+1.0\*ln(1+arcsin(T10(x31)))+1.0\*ln(1+arcsin(T11(x31)))+1.0\*ln(1+arcsin(T12(x31)))+1.0\*ln(1+arcsin(T13(x31)))+1.0\*ln(1+arcsin(T14(x31)))+1.0\*ln(1+arcsin(T15(x31))))-1

psi110=exp(+1.0\*ln(1+arcsin(T0(x10)))+1.0\*ln(1+arcsin(T1(x10)))+1.0\*ln(1+arcsin(T2(x10)))+1.0\*ln(1+arcsin(T3(x10)))+1.0\*ln(1+arcsin(T4(x10)))+1.0\*ln(1+arcsin(T5(x10)))+1.0\*ln(1+arcsin(T6(x10)))+1.0\*ln(1+arcsin(T7(x10)))+1.0\*ln(1+arcsin(T8(x10)))+1.0\*ln(1+arcsin(T9(x10)))+1.0\*ln(1+arcsin(T10(x10)))+1.0\*ln(1+arcsin(T11(x10)))+1.0\*ln(1+arcsin(T12(x10)))+1.0\*ln(1+arcsin(T13(x10)))+1.0\*ln(1+arcsin(T14(x10)))+1.0\*ln(1+arcsin(T15(x10))))-1

psi111=exp(+1.0\*ln(1+arcsin(T0(x11)))+1.0\*ln(1+arcsin(T1(x11)))+1.0\*ln(1+arcsin(T2(x11)))+1.0\*ln(1+arcsin(T3(x11)))+1.0\*ln(1+arcsin(T4(x11)))+1.0\*ln(1+arcsin(T5(x11)))+1.0\*ln(1+arcsin(T6(x11)))+1.0\*ln(1+arcsin(T7(x11)))+1.0\*ln(1+arcsin(T8(x11)))+1.0\*ln(1+arcsin(T9(x11)))+1.0\*ln(1+arcsin(T10(x11)))+1.0\*ln(1+arcsin(T11(x11)))+1.0\*ln(1+arcsin(T12(x11)))+1.0\*ln(1+arcsin(T13(x11)))+1.0\*ln(1+arcsin(T14(x11)))+1.0\*ln(1+arcsin(T15(x11))))-1

psi120=exp(+1.0\*ln(1+arcsin(T0(x20)))+1.0\*ln(1+arcsin(T1(x20)))+1.0\*ln(1+arcsin(T2(x20)))+1.0\*ln(1+arcsin(T3(x20)))+1.0\*ln(1+arcsin(T4(x20)))+1.0\*ln(1+arcsin(T5(x20)))+1.0\*ln(1+arcsin(T6(x20)))+1.0\*ln(1+arcsin(T7(x20)))+1.0\*ln(1+arcsin(T8(x20)))+1.0\*ln(1+arcsin(T9(x20)))+1.0\*ln(1+arcsin(T10(x20)))+1.0\*ln(1+arcsin(T11(x20)))+1.0\*ln(1+arcsin(T12(x20)))+1.0\*ln(1+arcsin(T13(x20)))+1.0\*ln(1+arcsin(T14(x20)))+1.0\*ln(1+arcsin(T15(x20))))-1

psi121=exp(+1.0\*ln(1+arcsin(T0(x21)))+1.0\*ln(1+arcsin(T1(x21)))+1.0\*ln(1+arcsin(T2(x21)))+1.0\*ln(1+arcsin(T3(x21)))+1.0\*ln(1+arcsin(T4(x21)))+1.0\*ln(1+arcsin(T5(x21)))+1.0\*ln(1+arcsin(T6(x21)))+1.0\*ln(1+arcsin(T7(x21)))+1.0\*ln(1+arcsin(T8(x21)))+1.0\*ln(1+arcsin(T9(x21)))+1.0\*ln(1+arcsin(T10(x21)))+1.0\*ln(1+arcsin(T11(x21)))+1.0\*ln(1+arcsin(T12(x21)))+1.0\*ln(1+arcsin(T13(x21)))+1.0\*ln(1+arcsin(T14(x21)))+1.0\*ln(1+arcsin(T15(x21))))-1

psi130=exp(+1.0\*ln(1+arcsin(T0(x30)))+1.0\*ln(1+arcsin(T1(x30)))+1.0\*ln(1+arcsin(T2(x30)))+1.0\*ln(1+arcsin(T3(x30)))+1.0\*ln(1+arcsin(T4(x30)))+1.0\*ln(1+arcsin(T5(x30)))+1.0\*ln(1+arcsin(T6(x30)))+1.0\*ln(1+arcsin(T7(x30)))+1.0\*ln(1+arcsin(T8(x30)))+1.0\*ln(1+arcsin(T9(x30)))+1.0\*ln(1+arcsin(T10(x30)))+1.0\*ln(1+arcsin(T11(x30)))+1.0\*ln(1+arcsin(T12(x30)))+1.0\*ln(1+arcsin(T13(x30)))+1.0\*ln(1+arcsin(T14(x30)))+1.0\*ln(1+arcsin(T15(x30))))-1

psi131=exp(+1.0\*ln(1+arcsin(T0(x31)))+1.0\*ln(1+arcsin(T1(x31)))+1.0\*ln(1+arcsin(T2(x31)))+1.0\*ln(1+arcsin(T3(x31)))+1.0\*ln(1+arcsin(T4(x31)))+1.0\*ln(1+arcsin(T5(x31)))+1.0\*ln(1+arcsin(T6(x31)))+1.0\*ln(1+arcsin(T7(x31)))+1.0\*ln(1+arcsin(T8(x31)))+1.0\*ln(1+arcsin(T9(x31)))+1.0\*ln(1+arcsin(T10(x31)))+1.0\*ln(1+arcsin(T11(x31)))+1.0\*ln(1+arcsin(T12(x31)))+1.0\*ln(1+arcsin(T13(x31)))+1.0\*ln(1+arcsin(T14(x31)))+1.0\*ln(1+arcsin(T15(x31))))-1

Ф01=+exp(+1.0001878492034755\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x10)))+1.0\*ln(1+arcsin(T1(x10)))+1.0\*ln(1+arcsin(T2(x10)))+1.0\*ln(1+arcsin(T3(x10)))+1.0\*ln(1+arcsin(T4(x10)))+1.0\*ln(1+arcsin(T5(x10)))+1.0\*ln(1+arcsin(T6(x10)))+1.0\*ln(1+arcsin(T7(x10)))+1.0\*ln(1+arcsin(T8(x10)))+1.0\*ln(1+arcsin(T9(x10)))+1.0\*ln(1+arcsin(T10(x10)))+1.0\*ln(1+arcsin(T11(x10)))+1.0\*ln(1+arcsin(T12(x10)))+1.0\*ln(1+arcsin(T13(x10)))+1.0\*ln(1+arcsin(T14(x10)))+1.0\*ln(1+arcsin(T15(x10))))-1)+1.0001881721842285\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x11)))+1.0\*ln(1+arcsin(T1(x11)))+1.0\*ln(1+arcsin(T2(x11)))+1.0\*ln(1+arcsin(T3(x11)))+1.0\*ln(1+arcsin(T4(x11)))+1.0\*ln(1+arcsin(T5(x11)))+1.0\*ln(1+arcsin(T6(x11)))+1.0\*ln(1+arcsin(T7(x11)))+1.0\*ln(1+arcsin(T8(x11)))+1.0\*ln(1+arcsin(T9(x11)))+1.0\*ln(1+arcsin(T10(x11)))+1.0\*ln(1+arcsin(T11(x11)))+1.0\*ln(1+arcsin(T12(x11)))+1.0\*ln(1+arcsin(T13(x11)))+1.0\*ln(1+arcsin(T14(x11)))+1.0\*ln(1+arcsin(T15(x11))))-1))-1

Ф02=+exp(+0.9999429087899347\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x20)))+1.0\*ln(1+arcsin(T1(x20)))+1.0\*ln(1+arcsin(T2(x20)))+1.0\*ln(1+arcsin(T3(x20)))+1.0\*ln(1+arcsin(T4(x20)))+1.0\*ln(1+arcsin(T5(x20)))+1.0\*ln(1+arcsin(T6(x20)))+1.0\*ln(1+arcsin(T7(x20)))+1.0\*ln(1+arcsin(T8(x20)))+1.0\*ln(1+arcsin(T9(x20)))+1.0\*ln(1+arcsin(T10(x20)))+1.0\*ln(1+arcsin(T11(x20)))+1.0\*ln(1+arcsin(T12(x20)))+1.0\*ln(1+arcsin(T13(x20)))+1.0\*ln(1+arcsin(T14(x20)))+1.0\*ln(1+arcsin(T15(x20))))-1)+0.9999429116182572\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x21)))+1.0\*ln(1+arcsin(T1(x21)))+1.0\*ln(1+arcsin(T2(x21)))+1.0\*ln(1+arcsin(T3(x21)))+1.0\*ln(1+arcsin(T4(x21)))+1.0\*ln(1+arcsin(T5(x21)))+1.0\*ln(1+arcsin(T6(x21)))+1.0\*ln(1+arcsin(T7(x21)))+1.0\*ln(1+arcsin(T8(x21)))+1.0\*ln(1+arcsin(T9(x21)))+1.0\*ln(1+arcsin(T10(x21)))+1.0\*ln(1+arcsin(T11(x21)))+1.0\*ln(1+arcsin(T12(x21)))+1.0\*ln(1+arcsin(T13(x21)))+1.0\*ln(1+arcsin(T14(x21)))+1.0\*ln(1+arcsin(T15(x21))))-1))-1

Ф03=+exp(+0.999981750091818\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x30)))+1.0\*ln(1+arcsin(T1(x30)))+1.0\*ln(1+arcsin(T2(x30)))+1.0\*ln(1+arcsin(T3(x30)))+1.0\*ln(1+arcsin(T4(x30)))+1.0\*ln(1+arcsin(T5(x30)))+1.0\*ln(1+arcsin(T6(x30)))+1.0\*ln(1+arcsin(T7(x30)))+1.0\*ln(1+arcsin(T8(x30)))+1.0\*ln(1+arcsin(T9(x30)))+1.0\*ln(1+arcsin(T10(x30)))+1.0\*ln(1+arcsin(T11(x30)))+1.0\*ln(1+arcsin(T12(x30)))+1.0\*ln(1+arcsin(T13(x30)))+1.0\*ln(1+arcsin(T14(x30)))+1.0\*ln(1+arcsin(T15(x30))))-1)+0.9999818634864125\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x31)))+1.0\*ln(1+arcsin(T1(x31)))+1.0\*ln(1+arcsin(T2(x31)))+1.0\*ln(1+arcsin(T3(x31)))+1.0\*ln(1+arcsin(T4(x31)))+1.0\*ln(1+arcsin(T5(x31)))+1.0\*ln(1+arcsin(T6(x31)))+1.0\*ln(1+arcsin(T7(x31)))+1.0\*ln(1+arcsin(T8(x31)))+1.0\*ln(1+arcsin(T9(x31)))+1.0\*ln(1+arcsin(T10(x31)))+1.0\*ln(1+arcsin(T11(x31)))+1.0\*ln(1+arcsin(T12(x31)))+1.0\*ln(1+arcsin(T13(x31)))+1.0\*ln(1+arcsin(T14(x31)))+1.0\*ln(1+arcsin(T15(x31))))-1))-1

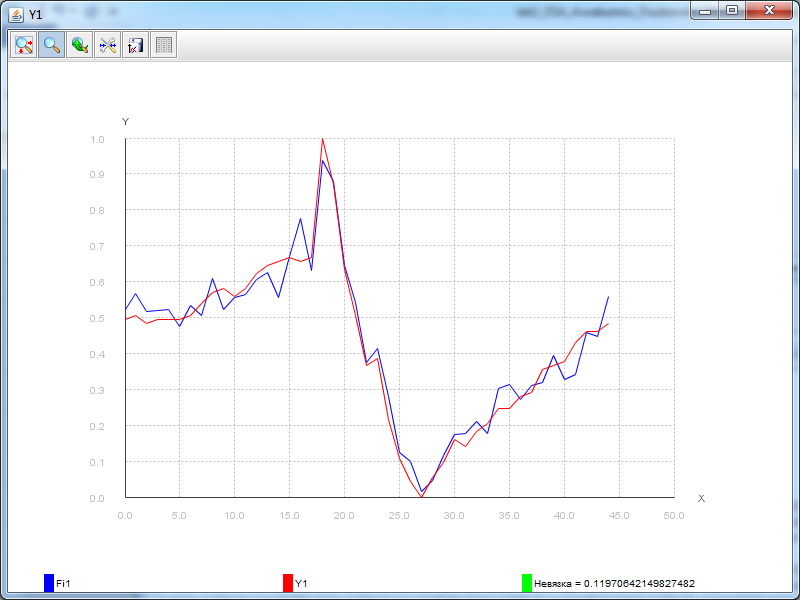
Ф11=+exp(+0.9994392496026523\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x10)))+1.0\*ln(1+arcsin(T1(x10)))+1.0\*ln(1+arcsin(T2(x10)))+1.0\*ln(1+arcsin(T3(x10)))+1.0\*ln(1+arcsin(T4(x10)))+1.0\*ln(1+arcsin(T5(x10)))+1.0\*ln(1+arcsin(T6(x10)))+1.0\*ln(1+arcsin(T7(x10)))+1.0\*ln(1+arcsin(T8(x10)))+1.0\*ln(1+arcsin(T9(x10)))+1.0\*ln(1+arcsin(T10(x10)))+1.0\*ln(1+arcsin(T11(x10)))+1.0\*ln(1+arcsin(T12(x10)))+1.0\*ln(1+arcsin(T13(x10)))+1.0\*ln(1+arcsin(T14(x10)))+1.0\*ln(1+arcsin(T15(x10))))-1)+0.9994395839351532\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x11)))+1.0\*ln(1+arcsin(T1(x11)))+1.0\*ln(1+arcsin(T2(x11)))+1.0\*ln(1+arcsin(T3(x11)))+1.0\*ln(1+arcsin(T4(x11)))+1.0\*ln(1+arcsin(T5(x11)))+1.0\*ln(1+arcsin(T6(x11)))+1.0\*ln(1+arcsin(T7(x11)))+1.0\*ln(1+arcsin(T8(x11)))+1.0\*ln(1+arcsin(T9(x11)))+1.0\*ln(1+arcsin(T10(x11)))+1.0\*ln(1+arcsin(T11(x11)))+1.0\*ln(1+arcsin(T12(x11)))+1.0\*ln(1+arcsin(T13(x11)))+1.0\*ln(1+arcsin(T14(x11)))+1.0\*ln(1+arcsin(T15(x11))))-1))-1

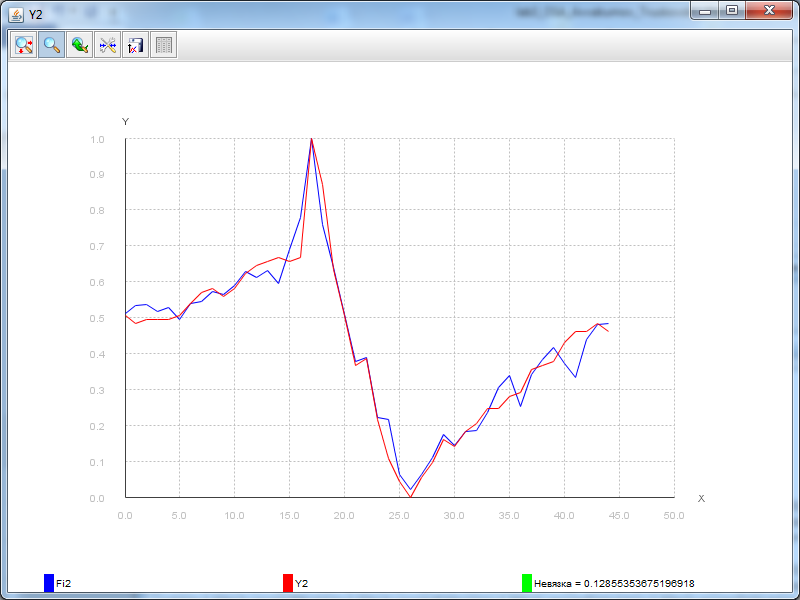
Ф12=+exp(+1.0001581290869817\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x20)))+1.0\*ln(1+arcsin(T1(x20)))+1.0\*ln(1+arcsin(T2(x20)))+1.0\*ln(1+arcsin(T3(x20)))+1.0\*ln(1+arcsin(T4(x20)))+1.0\*ln(1+arcsin(T5(x20)))+1.0\*ln(1+arcsin(T6(x20)))+1.0\*ln(1+arcsin(T7(x20)))+1.0\*ln(1+arcsin(T8(x20)))+1.0\*ln(1+arcsin(T9(x20)))+1.0\*ln(1+arcsin(T10(x20)))+1.0\*ln(1+arcsin(T11(x20)))+1.0\*ln(1+arcsin(T12(x20)))+1.0\*ln(1+arcsin(T13(x20)))+1.0\*ln(1+arcsin(T14(x20)))+1.0\*ln(1+arcsin(T15(x20))))-1)+1.0001585353874989\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x21)))+1.0\*ln(1+arcsin(T1(x21)))+1.0\*ln(1+arcsin(T2(x21)))+1.0\*ln(1+arcsin(T3(x21)))+1.0\*ln(1+arcsin(T4(x21)))+1.0\*ln(1+arcsin(T5(x21)))+1.0\*ln(1+arcsin(T6(x21)))+1.0\*ln(1+arcsin(T7(x21)))+1.0\*ln(1+arcsin(T8(x21)))+1.0\*ln(1+arcsin(T9(x21)))+1.0\*ln(1+arcsin(T10(x21)))+1.0\*ln(1+arcsin(T11(x21)))+1.0\*ln(1+arcsin(T12(x21)))+1.0\*ln(1+arcsin(T13(x21)))+1.0\*ln(1+arcsin(T14(x21)))+1.0\*ln(1+arcsin(T15(x21))))-1))-1

Ф13=+exp(+1.000247696761328\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x30)))+1.0\*ln(1+arcsin(T1(x30)))+1.0\*ln(1+arcsin(T2(x30)))+1.0\*ln(1+arcsin(T3(x30)))+1.0\*ln(1+arcsin(T4(x30)))+1.0\*ln(1+arcsin(T5(x30)))+1.0\*ln(1+arcsin(T6(x30)))+1.0\*ln(1+arcsin(T7(x30)))+1.0\*ln(1+arcsin(T8(x30)))+1.0\*ln(1+arcsin(T9(x30)))+1.0\*ln(1+arcsin(T10(x30)))+1.0\*ln(1+arcsin(T11(x30)))+1.0\*ln(1+arcsin(T12(x30)))+1.0\*ln(1+arcsin(T13(x30)))+1.0\*ln(1+arcsin(T14(x30)))+1.0\*ln(1+arcsin(T15(x30))))-1)+1.000247689092148\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x31)))+1.0\*ln(1+arcsin(T1(x31)))+1.0\*ln(1+arcsin(T2(x31)))+1.0\*ln(1+arcsin(T3(x31)))+1.0\*ln(1+arcsin(T4(x31)))+1.0\*ln(1+arcsin(T5(x31)))+1.0\*ln(1+arcsin(T6(x31)))+1.0\*ln(1+arcsin(T7(x31)))+1.0\*ln(1+arcsin(T8(x31)))+1.0\*ln(1+arcsin(T9(x31)))+1.0\*ln(1+arcsin(T10(x31)))+1.0\*ln(1+arcsin(T11(x31)))+1.0\*ln(1+arcsin(T12(x31)))+1.0\*ln(1+arcsin(T13(x31)))+1.0\*ln(1+arcsin(T14(x31)))+1.0\*ln(1+arcsin(T15(x31))))-1))-1

Ф0=sin(exp(+0.33585860746025153\*ln(1++exp(+1.0001878492034755\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x10)))+1.0\*ln(1+arcsin(T1(x10)))+1.0\*ln(1+arcsin(T2(x10)))+1.0\*ln(1+arcsin(T3(x10)))+1.0\*ln(1+arcsin(T4(x10)))+1.0\*ln(1+arcsin(T5(x10)))+1.0\*ln(1+arcsin(T6(x10)))+1.0\*ln(1+arcsin(T7(x10)))+1.0\*ln(1+arcsin(T8(x10)))+1.0\*ln(1+arcsin(T9(x10)))+1.0\*ln(1+arcsin(T10(x10)))+1.0\*ln(1+arcsin(T11(x10)))+1.0\*ln(1+arcsin(T12(x10)))+1.0\*ln(1+arcsin(T13(x10)))+1.0\*ln(1+arcsin(T14(x10)))+1.0\*ln(1+arcsin(T15(x10))))-1)+1.0001881721842285\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x11)))+1.0\*ln(1+arcsin(T1(x11)))+1.0\*ln(1+arcsin(T2(x11)))+1.0\*ln(1+arcsin(T3(x11)))+1.0\*ln(1+arcsin(T4(x11)))+1.0\*ln(1+arcsin(T5(x11)))+1.0\*ln(1+arcsin(T6(x11)))+1.0\*ln(1+arcsin(T7(x11)))+1.0\*ln(1+arcsin(T8(x11)))+1.0\*ln(1+arcsin(T9(x11)))+1.0\*ln(1+arcsin(T10(x11)))+1.0\*ln(1+arcsin(T11(x11)))+1.0\*ln(1+arcsin(T12(x11)))+1.0\*ln(1+arcsin(T13(x11)))+1.0\*ln(1+arcsin(T14(x11)))+1.0\*ln(1+arcsin(T15(x11))))-1))-1)+0.3368901701808016\*ln(1++exp(+0.9999429087899347\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x20)))+1.0\*ln(1+arcsin(T1(x20)))+1.0\*ln(1+arcsin(T2(x20)))+1.0\*ln(1+arcsin(T3(x20)))+1.0\*ln(1+arcsin(T4(x20)))+1.0\*ln(1+arcsin(T5(x20)))+1.0\*ln(1+arcsin(T6(x20)))+1.0\*ln(1+arcsin(T7(x20)))+1.0\*ln(1+arcsin(T8(x20)))+1.0\*ln(1+arcsin(T9(x20)))+1.0\*ln(1+arcsin(T10(x20)))+1.0\*ln(1+arcsin(T11(x20)))+1.0\*ln(1+arcsin(T12(x20)))+1.0\*ln(1+arcsin(T13(x20)))+1.0\*ln(1+arcsin(T14(x20)))+1.0\*ln(1+arcsin(T15(x20))))-1)+0.9999429116182572\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x21)))+1.0\*ln(1+arcsin(T1(x21)))+1.0\*ln(1+arcsin(T2(x21)))+1.0\*ln(1+arcsin(T3(x21)))+1.0\*ln(1+arcsin(T4(x21)))+1.0\*ln(1+arcsin(T5(x21)))+1.0\*ln(1+arcsin(T6(x21)))+1.0\*ln(1+arcsin(T7(x21)))+1.0\*ln(1+arcsin(T8(x21)))+1.0\*ln(1+arcsin(T9(x21)))+1.0\*ln(1+arcsin(T10(x21)))+1.0\*ln(1+arcsin(T11(x21)))+1.0\*ln(1+arcsin(T12(x21)))+1.0\*ln(1+arcsin(T13(x21)))+1.0\*ln(1+arcsin(T14(x21)))+1.0\*ln(1+arcsin(T15(x21))))-1))-1)+0.3392121503563523\*ln(1++exp(+0.999981750091818\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x30)))+1.0\*ln(1+arcsin(T1(x30)))+1.0\*ln(1+arcsin(T2(x30)))+1.0\*ln(1+arcsin(T3(x30)))+1.0\*ln(1+arcsin(T4(x30)))+1.0\*ln(1+arcsin(T5(x30)))+1.0\*ln(1+arcsin(T6(x30)))+1.0\*ln(1+arcsin(T7(x30)))+1.0\*ln(1+arcsin(T8(x30)))+1.0\*ln(1+arcsin(T9(x30)))+1.0\*ln(1+arcsin(T10(x30)))+1.0\*ln(1+arcsin(T11(x30)))+1.0\*ln(1+arcsin(T12(x30)))+1.0\*ln(1+arcsin(T13(x30)))+1.0\*ln(1+arcsin(T14(x30)))+1.0\*ln(1+arcsin(T15(x30))))-1)+0.9999818634864125\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x31)))+1.0\*ln(1+arcsin(T1(x31)))+1.0\*ln(1+arcsin(T2(x31)))+1.0\*ln(1+arcsin(T3(x31)))+1.0\*ln(1+arcsin(T4(x31)))+1.0\*ln(1+arcsin(T5(x31)))+1.0\*ln(1+arcsin(T6(x31)))+1.0\*ln(1+arcsin(T7(x31)))+1.0\*ln(1+arcsin(T8(x31)))+1.0\*ln(1+arcsin(T9(x31)))+1.0\*ln(1+arcsin(T10(x31)))+1.0\*ln(1+arcsin(T11(x31)))+1.0\*ln(1+arcsin(T12(x31)))+1.0\*ln(1+arcsin(T13(x31)))+1.0\*ln(1+arcsin(T14(x31)))+1.0\*ln(1+arcsin(T15(x31))))-1))-1))-1)

Ф1=sin(exp(+0.048347736657690536\*ln(1++exp(+0.9994392496026523\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x10)))+1.0\*ln(1+arcsin(T1(x10)))+1.0\*ln(1+arcsin(T2(x10)))+1.0\*ln(1+arcsin(T3(x10)))+1.0\*ln(1+arcsin(T4(x10)))+1.0\*ln(1+arcsin(T5(x10)))+1.0\*ln(1+arcsin(T6(x10)))+1.0\*ln(1+arcsin(T7(x10)))+1.0\*ln(1+arcsin(T8(x10)))+1.0\*ln(1+arcsin(T9(x10)))+1.0\*ln(1+arcsin(T10(x10)))+1.0\*ln(1+arcsin(T11(x10)))+1.0\*ln(1+arcsin(T12(x10)))+1.0\*ln(1+arcsin(T13(x10)))+1.0\*ln(1+arcsin(T14(x10)))+1.0\*ln(1+arcsin(T15(x10))))-1)+0.9994395839351532\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x11)))+1.0\*ln(1+arcsin(T1(x11)))+1.0\*ln(1+arcsin(T2(x11)))+1.0\*ln(1+arcsin(T3(x11)))+1.0\*ln(1+arcsin(T4(x11)))+1.0\*ln(1+arcsin(T5(x11)))+1.0\*ln(1+arcsin(T6(x11)))+1.0\*ln(1+arcsin(T7(x11)))+1.0\*ln(1+arcsin(T8(x11)))+1.0\*ln(1+arcsin(T9(x11)))+1.0\*ln(1+arcsin(T10(x11)))+1.0\*ln(1+arcsin(T11(x11)))+1.0\*ln(1+arcsin(T12(x11)))+1.0\*ln(1+arcsin(T13(x11)))+1.0\*ln(1+arcsin(T14(x11)))+1.0\*ln(1+arcsin(T15(x11))))-1))-1)+0.4087868589012239\*ln(1++exp(+1.0001581290869817\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x20)))+1.0\*ln(1+arcsin(T1(x20)))+1.0\*ln(1+arcsin(T2(x20)))+1.0\*ln(1+arcsin(T3(x20)))+1.0\*ln(1+arcsin(T4(x20)))+1.0\*ln(1+arcsin(T5(x20)))+1.0\*ln(1+arcsin(T6(x20)))+1.0\*ln(1+arcsin(T7(x20)))+1.0\*ln(1+arcsin(T8(x20)))+1.0\*ln(1+arcsin(T9(x20)))+1.0\*ln(1+arcsin(T10(x20)))+1.0\*ln(1+arcsin(T11(x20)))+1.0\*ln(1+arcsin(T12(x20)))+1.0\*ln(1+arcsin(T13(x20)))+1.0\*ln(1+arcsin(T14(x20)))+1.0\*ln(1+arcsin(T15(x20))))-1)+1.0001585353874989\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x21)))+1.0\*ln(1+arcsin(T1(x21)))+1.0\*ln(1+arcsin(T2(x21)))+1.0\*ln(1+arcsin(T3(x21)))+1.0\*ln(1+arcsin(T4(x21)))+1.0\*ln(1+arcsin(T5(x21)))+1.0\*ln(1+arcsin(T6(x21)))+1.0\*ln(1+arcsin(T7(x21)))+1.0\*ln(1+arcsin(T8(x21)))+1.0\*ln(1+arcsin(T9(x21)))+1.0\*ln(1+arcsin(T10(x21)))+1.0\*ln(1+arcsin(T11(x21)))+1.0\*ln(1+arcsin(T12(x21)))+1.0\*ln(1+arcsin(T13(x21)))+1.0\*ln(1+arcsin(T14(x21)))+1.0\*ln(1+arcsin(T15(x21))))-1))-1)+0.5495108376737448\*ln(1++exp(+1.000247696761328\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x30)))+1.0\*ln(1+arcsin(T1(x30)))+1.0\*ln(1+arcsin(T2(x30)))+1.0\*ln(1+arcsin(T3(x30)))+1.0\*ln(1+arcsin(T4(x30)))+1.0\*ln(1+arcsin(T5(x30)))+1.0\*ln(1+arcsin(T6(x30)))+1.0\*ln(1+arcsin(T7(x30)))+1.0\*ln(1+arcsin(T8(x30)))+1.0\*ln(1+arcsin(T9(x30)))+1.0\*ln(1+arcsin(T10(x30)))+1.0\*ln(1+arcsin(T11(x30)))+1.0\*ln(1+arcsin(T12(x30)))+1.0\*ln(1+arcsin(T13(x30)))+1.0\*ln(1+arcsin(T14(x30)))+1.0\*ln(1+arcsin(T15(x30))))-1)+1.000247689092148\*ln(1+exp(+1.0\*ln(1+arcsin(T0(x31)))+1.0\*ln(1+arcsin(T1(x31)))+1.0\*ln(1+arcsin(T2(x31)))+1.0\*ln(1+arcsin(T3(x31)))+1.0\*ln(1+arcsin(T4(x31)))+1.0\*ln(1+arcsin(T5(x31)))+1.0\*ln(1+arcsin(T6(x31)))+1.0\*ln(1+arcsin(T7(x31)))+1.0\*ln(1+arcsin(T8(x31)))+1.0\*ln(1+arcsin(T9(x31)))+1.0\*ln(1+arcsin(T10(x31)))+1.0\*ln(1+arcsin(T11(x31)))+1.0\*ln(1+arcsin(T12(x31)))+1.0\*ln(1+arcsin(T13(x31)))+1.0\*ln(1+arcsin(T14(x31)))+1.0\*ln(1+arcsin(T15(x31))))-1))-1))-1)





1. *Провести обзор и привести имеющуюся литературу по данному вопросу.*   
     
   **1**. Червоненкис А.Я. Компьютерный анализ данных, 2009.

Главным преимуществом этой книги является очень понятное изложение сложных методов, часто используемых в компьютерных программах статистического анализа данных.  
Книга А. Червоненкиса — это первый полный научный труд по теории машинного обучения, главной проблемой которого ставится именно обоснование качества обучающихся моделей. Проблема обоснования для автора является источником поиска новых путей развития теории обучающихся машин.

2. Яремко Н.Н. Проблема восстановления функциональных зависимостей в задачах интерпретации косвенных результатов наблюдения. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XI Междунар. научно-техн. конф. – Пенза: ПДЗ, 2011. – С. 54‑56.

Предлагается стохастический вариант задачи восстановления функциональных зависимостей в классе сплайнов с точками сопряжения.

# 3. Recovering time-varying networks of dependencies in social and biological studies. Published online before print July 1, 2009, doi: 10.1073/pnas.0901910106 ,PNAS July 21, 2009 vol. 106 no. 29 11878-11883

A plausible representation of the relational information among entities in dynamic systems such as a living cell or a social community is a stochastic network that is topologically rewiring and semantically evolving over time. Although there is a rich literature in modeling static or temporally invariant networks, little has been done toward recovering the network structure when the networks are not observable in a dynamic context. In this article, we present a machine learning method called TESLA, which builds on a temporally smoothed l1-regularized logistic regression formalism that can be cast as a standard convex-optimization problem and solved efficiently by using generic solvers scalable to large networks. We report promising results on recovering simulated time-varying networks and on reverse engineering the latent sequence of temporally rewiring political and academic social networks from longitudinal data, and the evolving gene networks over >4,000 genes during the life cycle of Drosophila melanogaster from a microarray time course at a resolution limited only by sample frequency.

Текст программы

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package lab3;

/\*\*

\*

\* @author Vova

\*/

public class Lagger implements Polinom {

public double func(int power, double x) {

switch (power) {

case 0:

return 1.0;

case 1:

return -x + 1;

default:

return ((2.0 \* (power - 1) + 1 - x) \* func(power - 1, x)

- (power - 1) \* func(power - 2, x))/(power);

}

}

public double[][] polinom(int power) {

double[][] pol = new double[power + 1][power + 1];

pol[0][0] = 1;

if (power > 0) {

pol[1][0] = 1;

pol[1][1] = -1;

}

for (int i = 2; i < power + 1; i++) {

pol[i][0] = -(i - 1) \* (i - 1) \* pol[i - 2][0] + (2 \* i - 1) \* pol[i - 1][0];

for (int j = 1; j < power + 1; j++) {

pol[i][j] = -pol[i - 1][j - 1] - (i - 1) \* (i - 1) \* pol[i - 2][j] + (2 \* i - 1) \* pol[i - 1][j];

}

}

return pol;

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package lab3;

/\*\*

\*

\* @author Vova

\*/

public class Lejandra implements Polinom {

@Override

public double func(int power, double x) {

switch (power) {

case 0:

return 1;

case 1:

return x;

default:

return ((2.0 \* (power - 1) + 1) \*x\* func(power - 1, x)

- (power - 1) \* func(power - 2, x)) / (power);

}

}

public double[][] polinom(int power) {

double[][] pol = new double[power + 1][power + 1];

pol[0][0] = 1;

if (power > 0) {

pol[1][0] = 0;

pol[1][1] = 1;

}

for (int i = 2; i < power + 1; i++) {

pol[i][0] = -(i - 1) \* pol[i - 2][0] / (double) i;

for (int j = 1; j < power + 1; j++) {

pol[i][j] = ((2 \* i - 1) \* pol[i - 1][j - 1]

- (i - 1) \* pol[i - 2][j]) / (double) i;

}

}

return pol;

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package lab3;

/\*\*

\*

\* @author Vova

\*/

public class Lejandr2 implements Polinom {

@Override

public double func(int power, double x) {

return f(power,2\*x-1);

}

private double f(int power, double x) {

switch (power) {

case 0:

return 1;

case 1:

return x;

default:

return ((2.0 \* (power - 1) + 1) \*x\* func(power - 1, x)

- (power - 1) \* func(power - 2, x)) / (power);

}

}

public double[][] polinom(int power) {

double[][] pol = new double[power + 1][power + 1];

pol[0][0] = 1;

if (power > 0) {

pol[1][0] = -1;

pol[1][1] = 2;

}

for (int i = 2; i < power + 1; i++) {

pol[i][0] = - (2 \* i - 1)\* pol[i-1][0]-(i - 1) \* pol[i - 2][0] / (double) i;

for (int j = 1; j < power + 1; j++) {

pol[i][j] = ((2 \* i - 1) \*(2\* pol[i - 1][j - 1]-pol[i-1][j])

- (i - 1) \* pol[i - 2][j]) / (double) i;

}

}

return pol;

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package lab3;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author Vova

\*/

public class Matr {

private double[][] x1;

private double[][] x2;

private double[][] x3;

public Matr(int n, int m, int n2, int m2, int n3, int m3) {

x1 = new double[n][m];

x2 = new double[n2][m2];

x3 = new double[n3][m3];

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

x1[i][j] = 1;

}

}

for (int i = 0; i < n2; i++) {

for (int j = 0; j < m2; j++) {

x2[i][j] = 1;

}

}

for (int i = 0; i < n3; i++) {

for (int j = 0; j < m3; j++) {

x3[i][j] = 1;

}

}

}

public Matr(int n, int m, int n2, int m2, int n3, int m3, double[] x) {

x1 = new double[n][m];

x2 = new double[n2][m2];

x3 = new double[n3][m3];

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

x1[i][j] = x[i \* m + j];

}

}

for (int i = 0; i < n2; i++) {

for (int j = 0; j < m2; j++) {

x2[i][j] = x[i \* m2 + j + n \* m];

}

}

for (int i = 0; i < n3; i++) {

for (int j = 0; j < m3; j++) {

x3[i][j] = x[i \* m3 + j + n \* m + n2 \* m2];

}

}

}

public Matr(double[][] x, double[][] y, double[][] z) {

x1 = x;

x2 = y;

x3 = z;

}

public Matr add(Matr a) {

double[][] x;

if (x1.length != 0) {

x = new double[x1.length][x1[0].length];

} else {

x = new double[0][0];

}

double[][] y;

double[][] z;

if (x2.length != 0) {

y = new double[x2.length][x2[0].length];

} else {

y = new double[0][0];

}

if (x3.length != 0) {

z = new double[x3.length][x3[0].length];

} else {

z = new double[0][0];

}

for (int i = 0; i < x.length; i++) {

for (int j = 0; j < x[0].length; j++) {

x[i][j] = x1[i][j] + a.x1[i][j];

}

}

for (int i = 0; i < x2.length; i++) {

for (int j = 0; j < x2[0].length; j++) {

y[i][j] = x2[i][j] + a.x2[i][j];

}

}

for (int i = 0; i < x3.length; i++) {

for (int j = 0; j < x3[0].length; j++) {

z[i][j] = x3[i][j] + a.x3[i][j];

}

}

return new Matr(x, y, z);

}

public Matr mult(double a) {

double[][] x;

if (x1.length != 0) {

x = new double[x1.length][x1[0].length];

} else {

x = new double[0][0];

}

double[][] y;

double[][] z;

if (x2.length != 0) {

y = new double[x2.length][x2[0].length];

} else {

y = new double[0][0];

}

if (x3.length != 0) {

z = new double[x3.length][x3[0].length];

} else {

z = new double[0][0];

}

for (int i = 0; i < x.length; i++) {

for (int j = 0; j < x[0].length; j++) {

x[i][j] = x1[i][j] \* a;

}

}

for (int i = 0; i < x2.length; i++) {

for (int j = 0; j < x2[0].length; j++) {

y[i][j] = x2[i][j] \* a;

}

}

for (int i = 0; i < x3.length; i++) {

for (int j = 0; j < x3[0].length; j++) {

z[i][j] = x3[i][j] \* a;

}

}

return new Matr(x, y, z);

}

public Matr addToX(double a, int xi, int xj) {

double[][] x;

if (x1.length != 0) {

x = new double[x1.length][x1[0].length];

} else {

x = new double[0][0];

}

double[][] y;

double[][] z;

if (x2.length != 0) {

y = new double[x2.length][x2[0].length];

} else {

y = new double[0][0];

}

if (x3.length != 0) {

z = new double[x3.length][x3[0].length];

} else {

z = new double[0][0];

}

for (int i = 0; i < x.length; i++) {

for (int j = 0; j < x[0].length; j++) {

x[i][j] = x1[i][j];

}

}

for (int i = 0; i < x2.length; i++) {

for (int j = 0; j < x2[0].length; j++) {

y[i][j] = x2[i][j];

}

}

for (int i = 0; i < x3.length; i++) {

for (int j = 0; j < x3[0].length; j++) {

z[i][j] = x3[i][j];

}

}

if (xi < x.length) {

x[xi][xj] += a;

}

return new Matr(x, y, z);

}

public Matr addToZ(double a, int xi, int xj) {

double[][] x;

if (x1.length != 0) {

x = new double[x1.length][x1[0].length];

} else {

x = new double[0][0];

}

double[][] y;

double[][] z;

if (x2.length != 0) {

y = new double[x2.length][x2[0].length];

} else {

y = new double[0][0];

}

if (x3.length != 0) {

z = new double[x3.length][x3[0].length];

} else {

z = new double[0][0];

}

for (int i = 0; i < x.length; i++) {

for (int j = 0; j < x[0].length; j++) {

x[i][j] = x1[i][j];

}

}

for (int i = 0; i < x2.length; i++) {

for (int j = 0; j < x2[0].length; j++) {

y[i][j] = x2[i][j];

}

}

for (int i = 0; i < x3.length; i++) {

for (int j = 0; j < x3[0].length; j++) {

z[i][j] = x3[i][j];

}

}

if (xi < z.length) {

z[xi][xj] += a;

}

return new Matr(x, y, z);

}

public Matr addToY(double a, int xi, int xj) {

double[][] x;

if (x1.length != 0) {

x = new double[x1.length][x1[0].length];

} else {

x = new double[0][0];

}

double[][] y;

double[][] z;

if (x2.length != 0) {

y = new double[x2.length][x2[0].length];

} else {

y = new double[0][0];

}

if (x3.length != 0) {

z = new double[x3.length][x3[0].length];

} else {

z = new double[0][0];

}

for (int i = 0; i < x.length; i++) {

for (int j = 0; j < x[0].length; j++) {

x[i][j] = x1[i][j];

}

}

for (int i = 0; i < x2.length; i++) {

for (int j = 0; j < x2[0].length; j++) {

y[i][j] = x2[i][j];

}

}

for (int i = 0; i < x3.length; i++) {

for (int j = 0; j < x3[0].length; j++) {

z[i][j] = x3[i][j];

}

}

if (xi < y.length) {

y[xi][xj] += a;

}

return new Matr(x, y, z);

}

public double[][] getX1() {

return x1;

}

public void setX1(double[][] x1) {

this.x1 = x1;

}

public double[][] getX2() {

return x2;

}

public void setX2(double[][] x2) {

this.x2 = x2;

}

public double[][] getX3() {

return x3;

}

public void setX3(double[][] x3) {

this.x3 = x3;

}

public double[][] getX(int s) throws Exception {

switch (s) {

case 1:

return x1;

case 2:

return x2;

case 3:

return x3;

default:

throw new Exception();

}

}

public double norm() {

double norm = 0;

for (int i = 0; i < x1.length; i++) {

for (int j = 0; j < x1[0].length; j++) {

norm += x1[i][j] \* x1[i][j];

}

}

for (int i = 0; i < x2.length; i++) {

for (int j = 0; j < x2[0].length; j++) {

norm += x2[i][j] \* x2[i][j];

}

}

for (int i = 0; i < x3.length; i++) {

for (int j = 0; j < x3[0].length; j++) {

norm += x3[i][j] \* x3[i][j];

}

}

return Math.sqrt(norm);

}

public String toString(String name) {

StringBuilder str = new StringBuilder(name);

for (int i = 1; i < 4; i++) {

double[][] x = null;

try {

x = getX(i);

} catch (Exception ex) {

Logger.getLogger(Matr.class.getName()).log(Level.SEVERE, null, ex);

}

for (int q = 0; q < x.length; q++) {

for (int j = 0; j < x[0].length; j++) {

str.append(String.format("%.4f \t",x[q][j]));

}

str.append("\r\n");

}

str.append("\r\n");

}

return str.toString();

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package lab3;

import Jama.Matrix;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

import java.util.Arrays;

import java.util.Collections;

import java.util.Locale;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import org.math.plot.Plot2DPanel;

/\*\*

\*

\* @author Vova

\*/

public class Work {

public static final double EPS = 1e-3;

private double[][][] x;

private double[][][] xN;

private double[][] y;

private double[][] yN;

private double[][] ln\_yN\_1;

private double[][] bq0;

private int m;

private int length;

private int[] p;

private int[] n;

private double[] xG;

private double[][] yG;

private double[][] yNG;

private double[] nevyazka;

private Polinom pol;

private File openFile;

private File resultFile;

private int chooseBQ0;

private double[][] maxX;

private double[][] minX;

private int chooseSolveOfLambda;

private StringBuilder result;

public Work(File openFile, File outputFile, int k, int length, int m,

int[] n, int[] p, Polinom pol, int chooseBQ0, int chooseSolveOfLambda) {

result = new StringBuilder();

x = new double[k][][];

xN = new double[k][][];

y = new double[m][length];

for (int i = 0; i < k; i++) {

x[i] = new double[n[i]][length];

}

this.length = length;

this.m = m;

this.p = p;

this.n = n;

this.pol = pol;

this.openFile = openFile;

this.resultFile = outputFile;

this.chooseBQ0 = chooseBQ0;

this.chooseSolveOfLambda = chooseSolveOfLambda;

xG = new double[length];

for (int i = 0; i < length; i++) {

xG[i] = i;

}

}

public void drawGraphNorm(int index) {

JFrame frame = new JFrame("Y" + (index + 1));

Plot2DPanel plot = new Plot2DPanel("South");

double[][] z, b;

z = yNG;

b = yN;

plot.addLinePlot("Fi" + (index + 1), xG, z[index]);

plot.addLinePlot("Y" + (index + 1), xG, b[index]);

double[] a = {0, 0};

plot.addLinePlot("Невязка = " + Double.toString(nevyazka[index]), a, a);

frame.add(plot);

frame.setVisible(true);

frame.pack();

frame.setBounds(0, 0, 800, 600);

}

public void drawGraphNenorm(int index) {

JFrame frame = new JFrame("Y" + (index + 1));

Plot2DPanel plot = new Plot2DPanel("South");

double[][] z, b;

z = yG;

b = y;

plot.addLinePlot("Fi" + (index + 1), xG, z[index]);

plot.addLinePlot("Y" + (index + 1), xG, b[index]);

double[] a = {0, 0};

plot.addLinePlot("Невязка = " + Double.toString(nevyazka[index]), a, a);

frame.add(plot);

frame.setVisible(true);

frame.pack();

frame.setBounds(0, 0, 800, 600);

}

public String getResult() {

return result.toString();

}

public void work() {

Scanner sc = null;

PrintWriter pw = null;

try {

sc = new Scanner(openFile);

sc.useLocale(Locale.US);

for (int i = 0; i < length; i++) {

for (int k = 0; k < x.length; k++) {

for (int j = 0; j < x[k].length; j++) {

x[k][j][i] = sc.nextDouble();

}

}

for (int j = 0; j < y.length; j++) {

y[j][i] = sc.nextDouble();

}

}

} catch (FileNotFoundException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

} finally {

sc.close();

}

//Normalization

for (int i = 0; i < x.length; i++) {

xN[i] = norm(x[i]);

}

yN = norm(y);

ln\_yN\_1 = ln\_y\_1(yN);

// Definition of bq0

// if (chooseBQ0 == 1) {

// //

bq0 = ln\_yN\_1;

// }

// if (chooseBQ0 == 2) {

// bq0 = bq0(0, yN);

//

// }

// if (chooseBQ0 == 3) {

// bq0 = bq01(0, yN);

//

// }

maxX = max(x);

minX = min(x);

Psi[] psi = new Psi[m];

// Find lambda

Matr lambda = null;

Matr v = null;

try {

pw = new PrintWriter(resultFile);

for (int i = 0; i < m; i++) {

// double[] lambda\_i\_1 = conjGradMethod(f1\_1(pol, new int[]{n[0], 0, 0}, new int[]{p[0], 0, 0}, xN[0], xN[1], xN[2]), bq0[i]);

// double[] lambda\_i\_2 = conjGradMethod(f1\_1(pol, new int[]{0, n[1], 0}, new int[]{0, p[1], 0}, xN[0], xN[1], xN[2]), bq0[i]);

// double[] lambda\_i\_3 = conjGradMethod(f1\_1(pol, new int[]{0, 0, n[2]}, new int[]{0, 0, p[2]}, xN[0], xN[1], xN[2]), bq0[i]);

// double[] lambda\_i = new double[lambda\_i\_1.length + lambda\_i\_2.length + lambda\_i\_3.length];

// System.arraycopy(lambda\_i\_1, 0, lambda\_i, 0, lambda\_i\_1.length);

// System.arraycopy(lambda\_i\_2, 0, lambda\_i, lambda\_i\_1.length, lambda\_i\_2.length);

// System.arraycopy(lambda\_i\_3, 0, lambda\_i, lambda\_i\_1.length + lambda\_i\_2.length, lambda\_i\_3.length);

lambda = new Matr(n[0], p[0] + 1, n[1], p[1] + 1, n[2], p[2] + 1);//, lambda\_i);

double[] v\_i\_1 = conjGradMethod(f1(pol,lambda, new int[]{n[0], 0, 0}, new int[]{p[0], 0, 0}, xN[0], xN[1], xN[2]), bq0[i]);

double[] v\_i\_2 = conjGradMethod(f1(pol, lambda,new int[]{0, n[1], 0}, new int[]{0, p[1], 0}, xN[0], xN[1], xN[2]), bq0[i]);

double[] v\_i\_3 = conjGradMethod(f1(pol,lambda, new int[]{0, 0, n[2]}, new int[]{0, 0, p[2]}, xN[0], xN[1], xN[2]), bq0[i]);

double[] v\_i = new double[v\_i\_1.length + v\_i\_2.length + v\_i\_3.length];

System.arraycopy(v\_i\_1, 0, v\_i, 0, v\_i\_1.length);

System.arraycopy(v\_i\_2, 0, v\_i, v\_i\_1.length, v\_i\_2.length);

System.arraycopy(v\_i\_3, 0, v\_i, v\_i\_1.length + v\_i\_2.length, v\_i\_3.length);

v = new Matr(n[0], p[0] + 1, n[1], p[1] + 1, n[2], p[2] + 1, v\_i);

psi[i] = new Psi(n[0], n[1], n[2], p[0], p[1], p[2], pol, lambda, v, maxX, minX);

result.append(v.toString("lambda " + i + ":\r\n"));

pw.println(v.toString("lambda " + i + ":\r\n"));

}

double[][] a1 = new double[m][];

double[][] a2 = new double[m][];

double[][] a3 = new double[m][];

for (int i = 0; i < m; i++) {

a1[i] = conjGradMethod(f2(psi, xN[0], 1, i), ln\_yN\_1[i]);

a2[i] = conjGradMethod(f2(psi, xN[1], 2, i), ln\_yN\_1[i]);

a3[i] = conjGradMethod(f2(psi, xN[2], 3, i), ln\_yN\_1[i]);}

Matr a = new Matr(a1, a2, a3);

result.append(a.toString("a\r\n"));

pw.println(a.toString("a\r\n"));

Fi fi = new Fi(n[0], n[1], n[2], psi, a);

double[][] c1 = new double[1][m];

double[][] c2 = new double[1][m];

double[][] c3 = new double[1][m];

for (int i = 0; i < m; i++) {

double[] c = conjGradMethod(f3(fi, xN[0], xN[1], xN[2], i), ln\_yN\_1[i]);

c1[0][i] = c[0];

c2[0][i] = c[1];

c3[0][i] = c[2];

}

Matr c = new Matr(c1, c2, c3);

result.append(c.toString("c\r\n"));

pw.println(c.toString("c\r\n"));

Fi\_i finalFunction = new Fi\_i(fi, c);

double[] maxj = new double[m];

double[] minj = new double[m];

for (int i = 0; i < y.length; i++) {

maxj[i] = y[i][0];

minj[i] = y[i][0];

for (int j = 1; j < y[i].length; j++) {

if (maxj[i] < y[i][j]) {

maxj[i] = y[i][j];

} else if (minj[i] > y[i][j]) {

minj[i] = y[i][j];

}

}

}

for (int i = 0; i < m; i++) {

for (int s = 1; s < 4; s++) {

for (int js = 0; js < n[s - 1]; js++) {

result.append("psi" + i + s + js + "=" + psi[i].toString(1, s, js) + "\n");

pw.println("psi" + i + s + js + "=" + psi[i].toString(1, s, js));

}

}

}

for (int i = 0; i < m; i++) {

for (int s = 1; s < 4; s++) {

result.append("Ф" + i + s + "=" + fi.toString(1, i, s) + "\n");

pw.println("Ф" + i + s + "=" + fi.toString(1, i, s));

}

}

for (int i = 0; i < m; i++) {

result.append("Ф" + i + "=" + finalFunction.toString(i) + "\n");

pw.println("Ф" + i + "=" + finalFunction.toString(i));

}

pw.println("В нормированном виде");

// for (int i = 0; i < m; i++) {

// pw.println("Ф" + i + "=" + finalFunction.toStringNorm(i));

//

// }

// pw.println("В ненормированном виде");

// for (int i = 0; i < m; i++) {

// pw.println("Ф" + i + "=" + finalFunction.toStringNenorm(i, maxj[i] - minj[i]) + "+" + minj);

//

// }

double[][] xForCalc = new double[x.length][];

for (int i = 0; i < x.length; i++) {

xForCalc[i] = new double[n[i]];

}

yG = new double[m][length];

yNG = new double[m][length];

nevyazka = new double[m];

pw.println("нормированные");

for (int j = 0; j < m; j++) {

pw.print("Y" + j + "\t\t");

}

pw.println();

for (int i = 0; i < length; i++) {

for (int k = 0; k < x.length; k++) {

for (int j = 0; j < n[k]; j++) {

xForCalc[k][j] = xN[k][j][i];

}

}

for (int j = 0; j < m; j++) {

yNG[j][i] = finalFunction.func(j, xForCalc[0], xForCalc[1], xForCalc[2]);

if (nevyazka[j] < Math.abs(yNG[j][i] - yN[j][i])) {

nevyazka[j] = Math.abs(yNG[j][i] - yN[j][i]);

};

pw.print(yNG[j][i] + "\t\t");

}

pw.println();

}

pw.println("без нормировки");

for (int j = 0; j < m; j++) {

pw.print("Y" + j + "\t\t");

}

pw.println();

for (int j = 0; j < y[0].length; j++) {

for (int i = 0; i < y.length; i++) {

yG[i][j] = yNG[i][j] \* (maxj[i] - minj[i]) + minj[i];

pw.print(yG[i][j] + "\t\t");

}

pw.println();

}

pw.println();

pw.println("Невязка");

for (int j = 0; j < y[0].length; j++) {

for (int i = 0; i < y.length; i++) {

pw.print(Math.abs(yNG[i][j] - yN[i][j]) + "\t\t");

}

pw.println();

}

} catch (FileNotFoundException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

} finally {

pw.close();

}

}

public void work1() {

Scanner sc = null;

PrintWriter pw = null;

try {

sc = new Scanner(openFile);

sc.useLocale(Locale.US);

for (int i = 0; i < length; i++) {

for (int k = 0; k < x.length; k++) {

for (int j = 0; j < x[k].length; j++) {

x[k][j][i] = sc.nextDouble();

}

}

for (int j = 0; j < y.length; j++) {

y[j][i] = sc.nextDouble();

}

}

} catch (FileNotFoundException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

} finally {

sc.close();

}

//Normalization

for (int i = 0; i < x.length; i++) {

xN[i] = norm(x[i]);

}

yN = norm(y);

ln\_yN\_1 = ln\_f\_y\_1(yN);

// Definition of bq0

// if (chooseBQ0 == 1) {

//

bq0 = ln\_yN\_1;

// }

// if (chooseBQ0 == 2) {

// bq0 = bq0(0, yN);

//

// }

// if (chooseBQ0 == 3) {

// bq0 = bq01(0, yN);

//

// }

maxX = max(x);

minX = min(x);

PsiOur[] psi = new PsiOur[m];

// Find lambda

Matr lambda = null;

Matr v = null;

try {

pw = new PrintWriter(resultFile);

for (int i = 0; i < m; i++) {

// double[] lambda\_i\_1 = conjGradMethod(f1\_1(pol, new int[]{n[0], 0, 0}, new int[]{p[0], 0, 0}, xN[0], xN[1], xN[2]), bq0[i]);

// double[] lambda\_i\_2 = conjGradMethod(f1\_1(pol, new int[]{0, n[1], 0}, new int[]{0, p[1], 0}, xN[0], xN[1], xN[2]), bq0[i]);

// double[] lambda\_i\_3 = conjGradMethod(f1\_1(pol, new int[]{0, 0, n[2]}, new int[]{0, 0, p[2]}, xN[0], xN[1], xN[2]), bq0[i]);

// double[] lambda\_i = new double[lambda\_i\_1.length + lambda\_i\_2.length + lambda\_i\_3.length];

// System.arraycopy(lambda\_i\_1, 0, lambda\_i, 0, lambda\_i\_1.length);

// System.arraycopy(lambda\_i\_2, 0, lambda\_i, lambda\_i\_1.length, lambda\_i\_2.length);

// System.arraycopy(lambda\_i\_3, 0, lambda\_i, lambda\_i\_1.length + lambda\_i\_2.length, lambda\_i\_3.length);

lambda = new Matr(n[0], p[0] + 1, n[1], p[1] + 1, n[2], p[2] + 1);//, lambda\_i);

double[] v\_i\_1 = conjGradMethod(f1\_our(pol, new int[]{n[0], 0, 0}, new int[]{p[0], 0, 0}, xN[0], xN[1], xN[2]), bq0[i]);

double[] v\_i\_2 = conjGradMethod(f1\_our(pol,new int[]{0, n[1], 0}, new int[]{0, p[1], 0}, xN[0], xN[1], xN[2]), bq0[i]);

double[] v\_i\_3 = conjGradMethod(f1\_our(pol, new int[]{0, 0, n[2]}, new int[]{0, 0, p[2]}, xN[0], xN[1], xN[2]), bq0[i]);

double[] v\_i = new double[v\_i\_1.length + v\_i\_2.length + v\_i\_3.length];

System.arraycopy(v\_i\_1, 0, v\_i, 0, v\_i\_1.length);

System.arraycopy(v\_i\_2, 0, v\_i, v\_i\_1.length, v\_i\_2.length);

System.arraycopy(v\_i\_3, 0, v\_i, v\_i\_1.length + v\_i\_2.length, v\_i\_3.length);

v = new Matr(n[0], p[0] + 1, n[1], p[1] + 1, n[2], p[2] + 1, v\_i);

psi[i] = new PsiOur(n[0], n[1], n[2], p[0], p[1], p[2], pol, lambda, v, maxX, minX);

result.append(v.toString("lambda " + i + ":\r\n"));

pw.println(v.toString("lambda " + i + ":\r\n"));

}

double[][] a1 = new double[m][];

double[][] a2 = new double[m][];

double[][] a3 = new double[m][];

for (int i = 0; i < m; i++) {

a1[i] = conjGradMethod(f2\_our(psi, xN[0], 1, i), ln\_yN\_1[i]);

a2[i] = conjGradMethod(f2\_our(psi, xN[1], 2, i), ln\_yN\_1[i]);

a3[i] = conjGradMethod(f2\_our(psi, xN[2], 3, i), ln\_yN\_1[i]);}

Matr a = new Matr(a1, a2, a3);

result.append(a.toString("a\r\n"));

pw.println(a.toString("a\r\n"));

FiOur fi = new FiOur(n[0], n[1], n[2], psi, a);

double[][] c1 = new double[1][m];

double[][] c2 = new double[1][m];

double[][] c3 = new double[1][m];

for (int i = 0; i < m; i++) {

double[] c = conjGradMethod(f3\_our(fi, xN[0], xN[1], xN[2], i), ln\_yN\_1[i]);

c1[0][i] = c[0];

c2[0][i] = c[1];

c3[0][i] = c[2];

}

Matr c = new Matr(c1, c2, c3);

result.append(c.toString("c\r\n"));

pw.println(c.toString("c\r\n"));

Fi\_iOur finalFunction = new Fi\_iOur(fi, c);

double[] maxj = new double[m];

double[] minj = new double[m];

for (int i = 0; i < y.length; i++) {

maxj[i] = y[i][0];

minj[i] = y[i][0];

for (int j = 1; j < y[i].length; j++) {

if (maxj[i] < y[i][j]) {

maxj[i] = y[i][j];

} else if (minj[i] > y[i][j]) {

minj[i] = y[i][j];

}

}

}

for (int i = 0; i < m; i++) {

for (int s = 1; s < 4; s++) {

for (int js = 0; js < n[s - 1]; js++) {

result.append("psi" + i + s + js + "=" + psi[i].toString(1, s, js) + "\n");

pw.println("psi" + i + s + js + "=" + psi[i].toString(1, s, js));

}

}

}

for (int i = 0; i < m; i++) {

for (int s = 1; s < 4; s++) {

result.append("Ф" + i + s + "=" + fi.toString(1, i, s) + "\n");

pw.println("Ф" + i + s + "=" + fi.toString(1, i, s));

}

}

for (int i = 0; i < m; i++) {

result.append("Ф" + i + "=" + finalFunction.toString(i) + "\n");

pw.println("Ф" + i + "=" + finalFunction.toString(i));

}

pw.println("В нормированном виде");

// for (int i = 0; i < m; i++) {

// pw.println("Ф" + i + "=" + finalFunction.toStringNorm(i));

//

// }

// pw.println("В ненормированном виде");

// for (int i = 0; i < m; i++) {

// pw.println("Ф" + i + "=" + finalFunction.toStringNenorm(i, maxj[i] - minj[i]) + "+" + minj);

//

// }

double[][] xForCalc = new double[x.length][];

for (int i = 0; i < x.length; i++) {

xForCalc[i] = new double[n[i]];

}

yG = new double[m][length];

yNG = new double[m][length];

nevyazka = new double[m];

pw.println("нормированные");

for (int j = 0; j < m; j++) {

pw.print("Y" + j + "\t\t");

}

pw.println();

for (int i = 0; i < length; i++) {

for (int k = 0; k < x.length; k++) {

for (int j = 0; j < n[k]; j++) {

xForCalc[k][j] = xN[k][j][i];

}

}

for (int j = 0; j < m; j++) {

yNG[j][i] = finalFunction.func(j, xForCalc[0], xForCalc[1], xForCalc[2]);

if (nevyazka[j] < Math.abs(yNG[j][i] - yN[j][i])) {

nevyazka[j] = Math.abs(yNG[j][i] - yN[j][i]);

};

pw.print(yNG[j][i] + "\t\t");

}

pw.println();

}

pw.println("без нормировки");

for (int j = 0; j < m; j++) {

pw.print("Y" + j + "\t\t");

}

pw.println();

for (int j = 0; j < y[0].length; j++) {

for (int i = 0; i < y.length; i++) {

yG[i][j] = yNG[i][j] \* (maxj[i] - minj[i]) + minj[i];

pw.print(yG[i][j] + "\t\t");

}

pw.println();

}

pw.println();

pw.println("Невязка");

for (int j = 0; j < y[0].length; j++) {

for (int i = 0; i < y.length; i++) {

pw.print(Math.abs(yNG[i][j] - yN[i][j]) + "\t\t");

}

pw.println();

}

} catch (FileNotFoundException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

} finally {

pw.close();

}

}

public double[][] norm(double[][] x) {

double[][] xn = new double[x.length][x[0].length];

for (int i = 0; i < x.length; i++) {

double maxj = x[i][0];

double minj = x[i][0];

for (int j = 1; j < x[i].length; j++) {

if (maxj < x[i][j]) {

maxj = x[i][j];

} else if (minj > x[i][j]) {

minj = x[i][j];

}

}

for (int j = 0; j < x[i].length; j++) {

xn[i][j] = (x[i][j] - minj) / (maxj - minj);

}

}

return xn;

}

private double[] maxXi(double[][] x) {

double[] max = new double[x.length];

for (int i = 0; i < x.length; i++) {

max[i] = x[i][0];

for (int j = 1; j < x[i].length; j++) {

if (max[i] < x[i][j]) {

max[i] = x[i][j];

}

}

}

return max;

}

public double[][] max(double[][][] x) {

double[][] max = new double[x.length][];

for (int i = 0; i < x.length; i++) {

max[i] = maxXi(x[i]);

}

return max;

}

private double[] minXi(double[][] x) {

double[] min = new double[x.length];

for (int i = 0; i < x.length; i++) {

min[i] = x[i][0];

for (int j = 1; j < x[i].length; j++) {

if (min[i] > x[i][j]) {

min[i] = x[i][j];

}

}

}

return min;

}

public double[][] min(double[][][] x) {

double[][] min = new double[x.length][];

for (int i = 0; i < x.length; i++) {

min[i] = minXi(x[i]);

}

return min;

}

public double[][] bq0(int index, double[][] y) {

double[][] bq0 = new double[y.length][y[0].length];

for (int i = 0; i < bq0[0].length; i++) {

double maxj = y[0][i];

double minj = y[0][i];

for (int j = 0; j < y.length; j++) {

if (maxj < y[j][i]) {

maxj = y[j][i];

} else if (minj > y[j][i]) {

minj = y[j][i];

}

}

for (int j = 0; j < bq0.length; j++) {

bq0[j][i] = Math.log(1 + (maxj + minj) / 2.0);

}

}

return bq0;

}

public double[][] bq01(int index, double[][] y) {

double[][] bq0 = new double[y.length][y[0].length];

for (int i = 0; i < bq0[0].length; i++) {

for (int j = 0; j < y.length; j++) {

bq0[j][i] = Math.log(1 + y[j][i]);

}

}

return bq0;

}

public double[] conjGradMethod(double[][] matrix, double[] vector) {

Matrix a = new Matrix(matrix);//, F.length, F[0].length);

Matrix b = new Matrix(vector, vector.length);

Matrix x = new Matrix(matrix.length, 1);

a = a.transpose();

// for (int i = 0; i < matrix.length; i++) {

// for (int j = 0; j < matrix[0].length; j++) {

// //if ((Double.isNaN(matrix[i][j]))) {

// if ((Double.isNaN(a.get(j, i)))) {

// System.out.println("hana " + i + " " + j);

// }

// }

// }

//if (Double.isNaN(a.norm2())) System.out.println("hana");

// if (Double.isNaN(b.norm2())) {

// System.out.println("hana");

// }

double rsNew = 0;

Matrix aTa = (a.transpose().times(a));

Matrix aTb = (a.transpose().times(b));

// if (Double.isNaN(aTb.norm2())) {

// System.out.println("hana");

// }

Matrix r = aTb.copy();//aTb.minus(aTa.times(x)).copy();

Matrix p = r.copy();

double rsOld = (r.transpose().times(r)).get(0, 0);

Matrix min = x;

double minNorm = a.times(min).minus(b).normInf();

for (int i = 0; i < 10e3; i++) {

Matrix Ap = aTa.times(p);

if ((p.transpose().times(Ap).get(0, 0)) == 0) {

System.out.println("asdasfdasfsfgdfg");

};

double alpha = rsOld / (p.transpose().times(Ap).get(0, 0));

//System.out.println((p.transpose().times(Ap).get(0, 0)));

if (Double.isNaN(alpha)) {

System.out.println(i + " " + rsOld + " " + (p.transpose().times(Ap).get(0, 0)));

break;

}

// if (Double.isNaN(alpha)) System.out.println("asdasfdasfsfgdfg");;

x = p.times(alpha).plus(x);

r = r.minus(Ap.times(alpha));

rsNew = (r.transpose().times(r)).get(0, 0);

double norm = a.times(x).minus(b).normInf();

if (norm < minNorm) {

min = x;

minNorm = norm;

}

if (Math.sqrt(rsNew) < 1e-10) {

break;

}

p = p.times(rsNew / rsOld).plus(r);

rsOld = rsNew;

}

double[] xRes = new double[matrix.length];

for (int i = 0; i < xRes.length; i++) {

xRes[i] = min.get(i, 0);

}

System.out.println("nevyazka " + minNorm);

return xRes;

}

public double[][] f1(Polinom pol, Matr lambda,int[] n, int[] p, double[][] x1, double[][] x2, double[][] x3) {

double[][] f1 = new double[n[0] \* p[0] + n[1] \* p[1] + n[2] \* p[2] + n[0] + n[1] + n[2]][x1[0].length];

int offset = 0;

for (int j = 0; j < n[0]; j++) {

for (int k = 0; k <= p[0]; k++) {

for (int q = 0; q < x1[0].length; q++) {

double temp = 1 + lambda.getX1()[j][k]\*pol.func(k, x1[j][q]);

if (temp < 0) {

System.out.println(temp);

System.out.println("");

}

f1[offset][q] = Math.log(Math.abs(1 + lambda.getX1()[j][k]\*pol.func(k, x1[j][q]))+10e-6);

}

offset++;

}

}

for (int j = 0; j < n[1]; j++) {

for (int k = 0; k <= p[1]; k++) {

for (int q = 0; q < x2[0].length; q++) {

f1[offset][q] = Math.log(Math.abs(1 + lambda.getX2()[j][k]\*pol.func(k, x2[j][q]))+10e-6);

}

offset++;

}

}

for (int j = 0; j < n[2]; j++) {

for (int k = 0; k <= p[2]; k++) {

for (int q = 0; q < x3[0].length; q++) {

f1[offset][q] = Math.log(Math.abs(1 + lambda.getX3()[j][k]\*pol.func(k, x3[j][q]))+10e-6);

}

offset++;

}

}

return f1;

}

public double[][] f1\_1(Polinom pol, int[] n, int[] p, double[][] x1, double[][] x2, double[][] x3) {

double[][] f1 = new double[n[0] \* p[0] + n[1] \* p[1] + n[2] \* p[2] + n[0] + n[1] + n[2]][x1[0].length];

int offset = 0;

for (int j = 0; j < n[0]; j++) {

for (int k = 0; k <= p[0]; k++) {

for (int q = 0; q < x1[0].length; q++) {

//double temp = 1 + pol.func(k, x1[j][q]);

//if (temp < 0) {

// System.out.println("dad");

//}

f1[offset][q] = pol.func(k, x1[j][q]);

}

offset++;

}

}

for (int j = 0; j < n[1]; j++) {

for (int k = 0; k <= p[1]; k++) {

for (int q = 0; q < x2[0].length; q++) {

f1[offset][q] = pol.func(k, x2[j][q]);

}

offset++;

}

}

for (int j = 0; j < n[2]; j++) {

for (int k = 0; k <= p[2]; k++) {

for (int q = 0; q < x3[0].length; q++) {

f1[offset][q] = pol.func(k, x3[j][q]);

}

offset++;

}

}

return f1;

}

public double[][] f2(Psi[] psi, double[][] xi, int s, int i) {

double[][] f2 = new double[xi.length][xi[0].length];

for (int k = 0; k < xi.length; k++) {

for (int j = 0; j < xi[0].length; j++) {

f2[k][j] = psi[i].ln\_psi\_s\_js\_1(s, k, xi[k][j]);

// if (Double.isInfinite(f2[k][j])) {

// System.out.println(f2[k][j]);

// System.out.println(psi[i].psi\_s\_js(s, k, xi[k][j]));

// System.out.println("");

// }

}

}

return f2;

}

public double[][] f3(Fi fi, double[][] x1, double[][] x2, double[][] x3, int i) {

double[][] f3 = new double[3][x1[0].length];

double[] x1q = new double[x1.length];

double[] x2q = new double[x2.length];

double[] x3q = new double[x3.length];

for (int q = 0; q < x1[0].length; q++) {

for (int k = 0; k < x1.length; k++) {

x1q[k] = x1[k][q];

}

for (int k = 0; k < x2.length; k++) {

x2q[k] = x2[k][q];

}

for (int k = 0; k < x3.length; k++) {

x3q[k] = x3[k][q];

}

f3[0][q] = fi.ln\_func\_1(i, 1, x1q);

f3[1][q] = fi.ln\_func\_1(i, 2, x2q);

f3[2][q] = fi.ln\_func\_1(i, 3, x3q);

}

return f3;

}

private double[][] ln\_y\_1(double[][] y) {

double[][] bq0 = new double[y.length][y[0].length];

for (int i = 0; i < bq0[0].length; i++) {

for (int j = 0; j < y.length; j++) {

bq0[j][i] = Math.log(1 + (y[j][i]));

}

}

return bq0;

}

private double[][] ln\_f\_y\_1(double[][] y) {

double[][] bq0 = new double[y.length][y[0].length];

for (int i = 0; i < bq0[0].length; i++) {

for (int j = 0; j < y.length; j++) {

bq0[j][i] = Math.log(1 + Func.func(y[j][i]));

}

}

return bq0;

}

public double[][] f1\_our(Polinom pol, int[] n, int[] p, double[][] x1, double[][] x2, double[][] x3) {

double[][] f1 = new double[n[0] \* p[0] + n[1] \* p[1] + n[2] \* p[2] + n[0] + n[1] + n[2]][x1[0].length];

int offset = 0;

for (int j = 0; j < n[0]; j++) {

for (int k = 0; k <= p[0]; k++) {

for (int q = 0; q < x1[0].length; q++) {

f1[offset][q] =Math.log(Math.abs(1+ Func.func(pol.func(k, x1[j][q])))+10e-6);

}

offset++;

}

}

for (int j = 0; j < n[1]; j++) {

for (int k = 0; k <= p[1]; k++) {

for (int q = 0; q < x2[0].length; q++) {

f1[offset][q] = Math.log(Math.abs(1+ Func.func(pol.func(k, x2[j][q])))+10e-6);

}

offset++;

}

}

for (int j = 0; j < n[2]; j++) {

for (int k = 0; k <= p[2]; k++) {

for (int q = 0; q < x3[0].length; q++) {

f1[offset][q] =Math.log(Math.abs(1+ Func.func(pol.func(k, x3[j][q])))+10e-6);

}

offset++;

}

}

return f1;

}

public double[][] f2\_our(PsiOur[] psi, double[][] xi, int s, int i) {

double[][] f2 = new double[xi.length][xi[0].length];

for (int k = 0; k < xi.length; k++) {

for (int j = 0; j < xi[0].length; j++) {

f2[k][j] = psi[i].ln\_psi\_s\_js\_1(s, k, xi[k][j]);

}

}

return f2;

}

public double[][] f3\_our(FiOur fi, double[][] x1, double[][] x2, double[][] x3, int i) {

double[][] f3 = new double[3][x1[0].length];

double[] x1q = new double[x1.length];

double[] x2q = new double[x2.length];

double[] x3q = new double[x3.length];

for (int q = 0; q < x1[0].length; q++) {

for (int k = 0; k < x1.length; k++) {

x1q[k] = x1[k][q];

}

for (int k = 0; k < x2.length; k++) {

x2q[k] = x2[k][q];

}

for (int k = 0; k < x3.length; k++) {

x3q[k] = x3[k][q];

}

f3[0][q] = fi.ln\_func\_1(i, 1, x1q);

f3[1][q] = fi.ln\_func\_1(i, 2, x2q);

f3[2][q] = fi.ln\_func\_1(i, 3, x3q);

}

return f3;

}

}

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package lab3;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Locale;

import java.util.Map;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.ComboBoxModel;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JFileChooser;

import javax.swing.JFrame;

import javax.swing.event.ListDataListener;

import org.math.plot.Plot2DPanel;

/\*\*

\*

\* @author Vova

\*/

public class MainFrame extends javax.swing.JFrame {

/\*\*

\* Creates new form MainFrame

\*/

public MainFrame() {

initComponents();

buttonGroup2.add(jRadioButton1);

buttonGroup2.add(jRadioButton2);

buttonGroup2.add(jRadioButton3);

buttonGroup2.add(jRadioButton4);

buttonGroup1.add(jRadioButton5);

buttonGroup1.add(jRadioButton6);

buttonGroup3.add(jRadioButton7);

buttonGroup3.add(jRadioButton8);

buttonGroup3.add(jRadioButton9);

jRadioButton9.setVisible(false);

jRadioButton1.setVisible(false);

//jPanel5.setVisible(false);

jPanel6.setVisible(false);

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

buttonGroup2 = new javax.swing.ButtonGroup();

buttonGroup1 = new javax.swing.ButtonGroup();

buttonGroup3 = new javax.swing.ButtonGroup();

jFileChooser1 = new javax.swing.JFileChooser();

jPanel3 = new javax.swing.JPanel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jTextField4 = new javax.swing.JTextField();

jTextField5 = new javax.swing.JTextField();

jTextField6 = new javax.swing.JTextField();

jTextField7 = new javax.swing.JTextField();

jPanel7 = new javax.swing.JPanel();

jButton3 = new javax.swing.JButton();

jComboBox1 = new javax.swing.JComboBox();

jCheckBox1 = new javax.swing.JCheckBox();

jPanel1 = new javax.swing.JPanel();

jLabel3 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jLabel1 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jTextField3 = new javax.swing.JTextField();

jButton2 = new javax.swing.JButton();

jPanel5 = new javax.swing.JPanel();

jRadioButton5 = new javax.swing.JRadioButton();

jRadioButton6 = new javax.swing.JRadioButton();

jPanel2 = new javax.swing.JPanel();

jRadioButton1 = new javax.swing.JRadioButton();

jRadioButton2 = new javax.swing.JRadioButton();

jRadioButton3 = new javax.swing.JRadioButton();

jRadioButton4 = new javax.swing.JRadioButton();

jButton4 = new javax.swing.JButton();

jPanel4 = new javax.swing.JPanel();

jLabel8 = new javax.swing.JLabel();

jTextField8 = new javax.swing.JTextField();

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

jTextField9 = new javax.swing.JTextField();

jTextField10 = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

jTextPane1 = new javax.swing.JTextPane();

jPanel6 = new javax.swing.JPanel();

jRadioButton7 = new javax.swing.JRadioButton();

jRadioButton8 = new javax.swing.JRadioButton();

jRadioButton9 = new javax.swing.JRadioButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("Lab\_3");

setPreferredSize(new java.awt.Dimension(720, 600));

setResizable(false);

jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder("Размерности"));

jLabel4.setText("X1");

jLabel5.setText("X2");

jLabel6.setText("X3");

jLabel7.setText("Y");

jTextField4.setText("2");

jTextField4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField4ActionPerformed(evt);

}

});

jTextField5.setText("2");

jTextField6.setText("2");

jTextField7.setText("2");

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);

jPanel3.setLayout(jPanel3Layout);

jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(jPanel3Layout.createSequentialGroup()

.addComponent(jLabel4)

.addGap(18, 18, 18)

.addComponent(jTextField4))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()

.addComponent(jLabel7)

.addGap(18, 18, 18)

.addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED\_SIZE, 43, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel3Layout.createSequentialGroup()

.addComponent(jLabel5)

.addGap(18, 18, 18)

.addComponent(jTextField5))

.addGroup(jPanel3Layout.createSequentialGroup()

.addComponent(jLabel6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED\_SIZE, 43, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel3Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {jLabel4, jLabel5, jLabel6, jLabel7});

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5)

.addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel6)

.addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel7)

.addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

);

jPanel7.setBorder(javax.swing.BorderFactory.createTitledBorder("График"));

jButton3.setText("Построить график");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Y1", "Y2", "Y3", "Y4" }));

jCheckBox1.setSelected(true);

jCheckBox1.setText("Нормировать");

javax.swing.GroupLayout jPanel7Layout = new javax.swing.GroupLayout(jPanel7);

jPanel7.setLayout(jPanel7Layout);

jPanel7Layout.setHorizontalGroup(

jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel7Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jCheckBox1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton3)

.addGap(18, 18, 18)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

jPanel7Layout.setVerticalGroup(

jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel7Layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addGroup(jPanel7Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton3)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jCheckBox1)))

);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Данные"));

jPanel1.setName(""); // NOI18N

jLabel3.setText("Входные данные");

jLabel2.setText("Размер выборки");

jTextField1.setText("45");

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jTextField2.setText("input.txt");

jLabel1.setText("Исходные данные");

jButton1.setText("Открыть");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jTextField3.setText("output.txt");

jButton2.setText("Выбрать");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel2)

.addComponent(jLabel3)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 95, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 41, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(105, 105, 105)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 81, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jTextField3)

.addGap(18, 18, 18)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 101, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(0, 19, Short.MAX\_VALUE)))

.addContainerGap())

);

jPanel1Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {jTextField2, jTextField3});

jPanel1Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {jButton1, jButton2});

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton1)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2))

.addGap(28, 28, 28))

);

jPanel5.setBorder(javax.swing.BorderFactory.createTitledBorder("Структура функций"));

jRadioButton5.setText("Предложенная");

jRadioButton5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jRadioButton5ActionPerformed(evt);

}

});

jRadioButton6.setSelected(true);

jRadioButton6.setText("Наша");

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);

jPanel5.setLayout(jPanel5Layout);

jPanel5Layout.setHorizontalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jRadioButton5)

.addComponent(jRadioButton6))

.addContainerGap(85, Short.MAX\_VALUE))

);

jPanel5Layout.setVerticalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup()

.addComponent(jRadioButton5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jRadioButton6))

);

jRadioButton5.getAccessibleContext().setAccessibleName("Предложенная");

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("Тип полинома"));

jRadioButton1.setText("Чебышев");

jRadioButton2.setSelected(true);

jRadioButton2.setText("Лаггер");

jRadioButton3.setText("Лежандр смещ");

jRadioButton4.setText("Лежандр");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jRadioButton1)

.addComponent(jRadioButton3)

.addComponent(jRadioButton2))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jRadioButton4)

.addContainerGap())

);

jPanel2Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {jRadioButton1, jRadioButton2, jRadioButton3, jRadioButton4});

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap(19, Short.MAX\_VALUE)

.addComponent(jRadioButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jRadioButton2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jRadioButton3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jRadioButton4))

);

jPanel2Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new java.awt.Component[] {jRadioButton1, jRadioButton2, jRadioButton3, jRadioButton4});

jButton4.setText("Запустить");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder("Степени полиномов"));

jLabel8.setText("при X1");

jTextField8.setText("7");

jLabel9.setText("при X2");

jLabel10.setText("при X3");

jTextField9.setText("7");

jTextField10.setText("7");

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(jPanel4Layout.createSequentialGroup()

.addComponent(jLabel10)

.addGap(18, 18, 18)

.addComponent(jTextField10))

.addGroup(jPanel4Layout.createSequentialGroup()

.addComponent(jLabel9)

.addGap(18, 18, 18)

.addComponent(jTextField9))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel4Layout.createSequentialGroup()

.addComponent(jLabel8)

.addGap(18, 18, 18)

.addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED\_SIZE, 37, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel8)

.addComponent(jTextField8, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel9)

.addComponent(jTextField9, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel10)

.addComponent(jTextField10, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jScrollPane1.setViewportView(jTextPane1);

jTextPane1.getAccessibleContext().setAccessibleName("Предложенная");

jPanel6.setBorder(javax.swing.BorderFactory.createTitledBorder("Подсчёт b0"));

jRadioButton7.setSelected(true);

jRadioButton7.setText("Нормированные Y");

jRadioButton8.setText("через минимум и максимум");

jRadioButton9.setText("Среднее значение Y");

jRadioButton9.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jRadioButton9ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel6Layout = new javax.swing.GroupLayout(jPanel6);

jPanel6.setLayout(jPanel6Layout);

jPanel6Layout.setHorizontalGroup(

jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel6Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jRadioButton7)

.addComponent(jRadioButton8)

.addComponent(jRadioButton9))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel6Layout.setVerticalGroup(

jPanel6Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel6Layout.createSequentialGroup()

.addComponent(jRadioButton7)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jRadioButton8)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jRadioButton9)

.addGap(0, 16, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(2, 2, 2)

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jPanel7, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPanel6, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(jScrollPane1)

.addGroup(layout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(jButton4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(78, 78, 78)

.addComponent(jPanel6, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(3, 3, 3)

.addComponent(jPanel7, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 161, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jButton4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 250, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(24, 24, 24))

);

jPanel1.getAccessibleContext().setAccessibleName("");

getAccessibleContext().setAccessibleDescription("");

pack();

}// </editor-fold>//GEN-END:initComponents

private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextField4ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextField4ActionPerformed

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton4ActionPerformed

jTextPane1.setText("");

try {

//Launch program

openFile.createNewFile();

resultFile.createNewFile();

//System.out.println(openFile.getAbsolutePath());

} catch (IOException ex) {

Logger.getLogger(MainFrame.class.getName()).log(Level.SEVERE, null, ex);

}

int length = Integer.parseInt(jTextField1.getText());

int n1 = Integer.parseInt(jTextField4.getText());

int n2 = Integer.parseInt(jTextField5.getText());

int n3 = Integer.parseInt(jTextField6.getText());

int p1 = Integer.parseInt(jTextField8.getText());

int p2 = Integer.parseInt(jTextField9.getText());

int p3 = Integer.parseInt(jTextField10.getText());

int m = Integer.parseInt(jTextField7.getText());

Polinom pol = null;

if (jRadioButton1.isSelected()) {

pol = new Chebyshev();

}

if (jRadioButton2.isSelected()) {

pol = new Lagger();

}

if (jRadioButton3.isSelected()) {

pol = new Lejandr2();

}

if (jRadioButton4.isSelected()) {

pol = new Lejandra();

}

int chooseBQ0 = 0;

if (jRadioButton7.isSelected()) {

chooseBQ0 = 1;

//

// bq0 = yN;

}

if (jRadioButton8.isSelected()) {

chooseBQ0 = 2;

//bq0 = Work.bq0(0, yN);

}

if (jRadioButton9.isSelected()) {

chooseBQ0 = 3;

//bq0 = Work.bq01(0, yN);

}

int chooseSolveLambda = 0;

if (jRadioButton5.isSelected()) {

chooseSolveLambda = 1;

} else if (jRadioButton6.isSelected()) {

chooseSolveLambda = 2;

}

String[] list = new String[m];

for (int i = 1; i <= m; i++) {

list[i - 1] = "Y" + i;

}

jComboBox1.setModel(new DefaultComboBoxModel(list));

work = new Work(openFile, resultFile, 3,length,m,

new int[]{n1, n2, n3}, new int[]{p1, p2, p3}, pol, chooseBQ0, chooseSolveLambda);

if (jRadioButton6.isSelected()) work.work1();

if (jRadioButton5.isSelected()) work.work();

write(work.getResult());

}//GEN-LAST:event\_jButton4ActionPerformed

private void write(String text) {

jTextPane1.setText(jTextPane1.getText() + text);

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton3ActionPerformed

if (!jCheckBox1.isSelected()) {

work.drawGraphNenorm(jComboBox1.getSelectedIndex());

} else {

work.drawGraphNorm(jComboBox1.getSelectedIndex());

}

// TODO add your handling code here:

}//GEN-LAST:event\_jButton3ActionPerformed

private void jRadioButton9ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jRadioButton9ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jRadioButton9ActionPerformed

private void jRadioButton5ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jRadioButton5ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jRadioButton5ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

JFileChooser fileopen = new JFileChooser();

int ret = fileopen.showDialog(null, "Выбрать файл");

if (ret == JFileChooser.APPROVE\_OPTION) {

resultFile = fileopen.getSelectedFile();

jTextField3.setText(openFile.getAbsolutePath());

} // TODO add your handling code here:

}//GEN-LAST:event\_jButton2ActionPerformed

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

JFileChooser fileopen = new JFileChooser();

int ret = fileopen.showDialog(null, "Открыть файл");

if (ret == JFileChooser.APPROVE\_OPTION) {

openFile = fileopen.getSelectedFile();

jTextField2.setText(openFile.getAbsolutePath());

}

//new OpenFile().setVisible(true);

//jFileChooser1.setVisible(true);// TODO add your handling code here:

}//GEN-LAST:event\_jButton1ActionPerformed

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jTextField1ActionPerformed

// TODO add your handling code here:

}//GEN-LAST:event\_jTextField1ActionPerformed

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new MainFrame().setVisible(true);

}

});

}

File openFile = new File("input.txt");

File resultFile = new File("output.txt");

private Work work;

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.ButtonGroup buttonGroup1;

private javax.swing.ButtonGroup buttonGroup2;

private javax.swing.ButtonGroup buttonGroup3;

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JCheckBox jCheckBox1;

private javax.swing.JComboBox jComboBox1;

private javax.swing.JFileChooser jFileChooser1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JPanel jPanel5;

private javax.swing.JPanel jPanel6;

private javax.swing.JPanel jPanel7;

private javax.swing.JRadioButton jRadioButton1;

private javax.swing.JRadioButton jRadioButton2;

private javax.swing.JRadioButton jRadioButton3;

private javax.swing.JRadioButton jRadioButton4;

private javax.swing.JRadioButton jRadioButton5;

private javax.swing.JRadioButton jRadioButton6;

private javax.swing.JRadioButton jRadioButton7;

private javax.swing.JRadioButton jRadioButton8;

private javax.swing.JRadioButton jRadioButton9;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField10;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField4;

private javax.swing.JTextField jTextField5;

private javax.swing.JTextField jTextField6;

private javax.swing.JTextField jTextField7;

private javax.swing.JTextField jTextField8;

private javax.swing.JTextField jTextField9;

private javax.swing.JTextPane jTextPane1;

// End of variables declaration//GEN-END:variables

}