Отчет по заданию №2

Первая краевая задача для одномерного стационарного уравнения теплопроводности с кусочно-непрерывными коэффициентами.

Вариант: 2
Выполнила: Бryндина А.А., 305 группа
Преподаватель: Есикова Н.Б.

Московский государственный университет имени М.В.Ломоносова
Факультет вычислительной математики и кибернетики

2020

Содержание

Постановка задачи

Нам необходимо найти решение обыкновенного дифференциального уравнения второго порядка с кусочно-непрерывными коэффициентами

$(k(x)u'(x))' - q(x)u(x) = -f(x),\ 0 < x < 1,$

и краевые условия:

$u(0) = u_1,\ u(1) = u_2$

$k(x) \geq c_1 > 0,\ q(x) \geq 0.$

$k(x), q(x), f(x)$ - кусочно-непрерывные функции, имеющие разрыв первого рода в точке x_0 .

В точке разрыва x_0 ставятся условия сопряжения:

$[u] = 0,\ [k'u] = 0$ при $x = x_0$

Вариант 2:

$x_0 = 0.525$ - точка разрыва

вид коэффициентов:

$k_1(x) = e^{-x^2},\ q_1(x) = x^2,\ f_1(x) = sin(x)$ при $x < x_0$,

$k_2(x) = x,\ q_2(x) = x^2,\ f_2(x) = sin(x)$ при $x > x_0$,

значения u_1, u_2:

$u_1 = 0,\ u_2 = 1,$

точность $\varepsilon = 0.01$

Аналитическое решение модельной задачи

Аналитическое решение задачи можно построить, если заменить функции $k(x), q(x), f(x)$ константами.

Рассмотрим модельную задачу с постоянными коэффициентами k, q, f:

$ku'' - qu = -f$

$u(0) = 0,\ u(1) = 1$

Тогда введем обозначения для коэффициентов:

$k = k_1(x_0) = k_1,\ q = q_1(x_0) = q_1,\ f = f_1(x_0) = f_1$ при $0 < x < x_0$,

$k = k_2(x_0) = k_2,\ q = q_2(x_0) = q_2,\ f = f_2(x_0) = f_2$ при $x_0 < x < 1$.

Общее решение задачи:

$u = C_1 e^{\lambda_1 x} + C_2 e^{-\lambda_1 x} + \mu_1,\ 0 < x < x_0$

$u = C_3 e^{\lambda_2 x} + C_4 e^{-\lambda_2 x} + \mu_2,\ x_0 < x < 1.$

μ_1, μ_2 - частное решение неоднородного уравнения,

где $\mu_1 = \frac{f_1}{q_1}, \mu_2 = \frac{f_2}{q_2}$

Как найти коэффициенты C_1, C_2, C_3, C_4 описано на стр.13(“Приложение”).

Численное решение задачи

На отрезке $0 \le x \le 1$ введем равномерную сетку

$\omega = \{x_i = ih,\ i = 0, 1, .., N,\ x_0 = 0, x_N = 1,\ h = 1/N\}$

Заменим следующей разностной задачей

$(a y_{\bar{x}})_{\hat{x},i} - d_i y_i = -\varphi_i\ ,\ 1 \le i \le N-1$

$y_0 = \mu_{1,}\ y_1 = \mu_2$

где $d_i = q(x_i),\ \varphi_i = f(x_i)$

Запишем разностную производную

$(a y_{\bar{x}})_{\hat{x},i} = 1/h(\ a_{i+1}\frac{y_{i+1} - y_i}{h}\ - a_i\frac{y_i - y_{i-1}}{h}\ )$

Задача записывается в виде системы

$\mathrm{C}_0 y_0 - B_0 y_1 = f_0,\ i\ = 0$

$-A_i y_{i-1} + C_i y_i - B_i y_{i+1} = f_i,\ 1 \le i \le N$

$-A_N y_{N-1} + C_N y_N = f_N,\ i = N$

Учтем,что

$B_0 = A_N =\ 0,\ C_0 = C_N = 1, f_0 = \mu_1, f_N = \mu_2, f_i = \varphi_i$

$A_i = \frac{k_i}{h^2}\ , B_i = \frac{k_{i+1}}{h^2}\ ,\ C_i = A_i + B_i + d_i,\ 1 \le i \le N-1$

Эта задача может быть решена методом правой прогонки

1) прямой ход прогонки

   $\alpha_{i+1} = \frac{B_i}{C_i - \alpha_i A_i}\ ,\ \beta_{i+1} = \frac{f_i + A_i \beta_i}{C_i - \alpha_i A_i}\ ,\ i = 1, 2, .., N-1$

2) обратный ход прогонки

   $y_N = \beta_{N+1}$

   $y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1},\ i = N-1, ..., 1, 0$

Пример работы программы

1)Пример работы программы для немодельной задачи

Start program! Enter N, please:

10

Model or no ? Enter m,please:

1

no model

x = 0	y_N[0] = 0	y_2N[0] = 0	delta = 0
x = 0.1	y_N[0.1] = 0.0947736	y_2N[0.1] = 0.0934821	delta = 0.00129152
x = 0.2	y_N[0.2] = 0.191404	y_2N[0.2] = 0.188315	delta = 0.00308955
x = 0.3	y_N[0.3] = 0.290899	y_2N[0.3] = 0.285459	delta = 0.00544047
x = 0.4	y_N[0.4] = 0.394448	y_2N[0.4] = 0.386035	delta = 0.00841264
x = 0.5	y_N[0.5] = 0.503558	y_2N[0.5] = 0.491451	delta = 0.0121071
x = 0.6	y_N[0.6] = 0.639291	y_2N[0.6] = 0.631853	delta = 0.00743793
x = 0.7	y_N[0.7] = 0.750855	y_2N[0.7] = 0.746413	delta = 0.00444157
x = 0.8	y_N[0.8] = 0.84502	y_2N[0.8] = 0.842582	delta = 0.00243745
x = 0.9	y_N[0.9] = 0.92676	y_2N[0.9] = 0.925723	delta = 0.00103681
x = 1	y_N[1] = 1	y_2N[1] = 1	delta = 0

max delta = 0.0121071 > 0.01

New N = 20

x = 0	y_N[0] = 0	y_2N[0] = 0	delta = 0
x = 0.05	y_N[0.05] = 0.0466285	y_2N[0.05] = 0.0463435	delta = 0.000284952
x = 0.1	y_N[0.1] = 0.0934821	y_2N[0.1] = 0.0928521	delta = 0.000629974
x = 0.15	y_N[0.15] = 0.140672	y_2N[0.15] = 0.139635	delta = 0.00103705
x = 0.2	y_N[0.2] = 0.188315	y_2N[0.2] = 0.186806	delta = 0.00150867
x = 0.25	y_N[0.25] = 0.236533	y_2N[0.25] = 0.234485	delta = 0.00204788
x = 0.3	y_N[0.3] = 0.285459	y_2N[0.3] = 0.2828	delta = 0.00265841
x = 0.35	y_N[0.35] = 0.335239	y_2N[0.35] = 0.331894	delta = 0.00334479
x = 0.4	y_N[0.4] = 0.386035	y_2N[0.4] = 0.381923	delta = 0.00411254
x = 0.45	y_N[0.45] = 0.438034	y_2N[0.45] = 0.433066	delta = 0.00496845
x = 0.5	y_N[0.5] = 0.491451	y_2N[0.5] = 0.48553	delta = 0.00592088
x = 0.55	y_N[0.55] = 0.565469	y_2N[0.55] = 0.560871	delta = 0.00459807
x = 0.6	y_N[0.6] = 0.631853	y_2N[0.6] = 0.628292	delta = 0.00356136
x = 0.65	y_N[0.65] = 0.691834	y_2N[0.65] = 0.689094	delta = 0.00273982
x = 0.7	y_N[0.7] = 0.746413	y_2N[0.7] = 0.744331	delta = 0.00208247
x = 0.75	y_N[0.75] = 0.796426	y_2N[0.75] = 0.794874	delta = 0.00155173
x = 0.8	y_N[0.8] = 0.842582	y_2N[0.8] = 0.841463	delta = 0.00111925
x = 0.85	y_N[0.85] = 0.8855	y_2N[0.85] = 0.884737	delta = 0.000763235
x = 0.9	y_N[0.9] = 0.925723	y_2N[0.9] = 0.925257	delta = 0.000466626
x = 0.95	y_N[0.95] = 0.963742	y_2N[0.95] = 0.963526	delta = 0.000215863
x = 1	y_N[1] = 1	y_2N[1] = 1	delta = 0

max delta = 0.00592088 < 0.01

2)Пример работы программы для модельной задачи

```
Start program! Enter N, please:
10
Model or no ? Enter m,please:
0
 model
x=0    u[0] =0              y[0] =0              delta = 0
x=0.1  u[0.1] =0.108396  y[0.1] =0.107654  delta = 0.000741417
x=0.2  u[0.2] =0.21058   y[0.2] =0.209097  delta = 0.00148365
x=0.3  u[0.3] =0.306925  y[0.3] =0.304696  delta = 0.0022295
x=0.4  u[0.4] =0.39778   y[0.4] =0.394798  delta = 0.00298178
x=0.5  u[0.5] =0.483475  y[0.5] =0.479731  delta = 0.00374333
x=0.6  u[0.6] =0.591173  y[0.6] =0.595508  delta = 0.00433441
x=0.7  u[0.7] =0.701637  y[0.7] =0.704864  delta = 0.00322747
x=0.8  u[0.8] =0.806234  y[0.8] =0.808374  delta = 0.00214004
x=0.9  u[0.9] =0.905514  y[0.9] =0.906581  delta = 0.00106617
x=1    u[1] =1              y[1] =1              delta = 0
max delta = 0.00433441 < 0.01
```

Исходный код программы

```
#include <stdio.h>
#include <iostream>
#include <cmath>
#include <math.h>
using namespace std;
double x0 = 0.525; // точка разрыва

void progonka(double* x, double* y, double* k, double* q, double* fi, int N, int m) {
    double* alfa; // прогоночные коэффициенты
    double* beta;
    double* a; // диагональные элементы матрицы
    double* b;
    double* c;
    double h = 1.0 / N; // шаг

    //cout << "h =" << h << '\n' ;
    for (int i = 0; i < N + 1; i++) {
        x[i] = 0.0;
        y[i] = 0.0;
        k[i] = 0.0;
        q[i] = 0.0;
        fi[i] = 0.0;
    }
    for (int i = 0; i < N + 1; i++) {
        x[i] = i * h;
    }
    for (int i = 0; i < N + 1; i++) {
        if (m == 1) { // немодельная задача
            if (x[i] < x0) {
                k[i] = exp(-x[i] * x[i]);
                q[i] = x[i] * x[i];
                fi[i] = sin(x[i]);
            }
            else {
                k[i] = x[i];
                q[i] = x[i] * x[i];
                fi[i] = sin(x[i]);
            }
        } else { // модельная задача
            if (x[i] < x0) {
                k[i] = exp(-x0 * x0);
                q[i] = x0 * x0;
                fi[i] = sin(x0);
            } else {
```

```
                k[i] = x0;
                q[i] = x0 * x0;
                fi[i] = sin(x0);
            }
        }
    }
    alfa = new double[N + 2];
    beta = new double[N + 2];
    a = new double[N + 2];
    b = new double[N + 2];
    c = new double[N + 2];
    for (int i = 1; i < N; i++) {
        a[i] = k[i] / (h * h);
        //cout << "a[" << i << "]=" << a[i] << " " << '\n';
        b[i] = k[i + 1] / (h * h);
        //cout << "b[" << i << "]=" << b[i] << " " << '\n';
        c[i] = k[i] / (h * h) + k[i + 1] / (h * h) + q[i];
        //cout << "c[" << i << "]=" << c[i] << " " << '\n';
    }
    b[0] = 0;
    a[N] = 0;
    c[0] = c[N] = 1;
    fi[0] = 0; fi[N] = 1;
    alfa[1] = 0.0;
    beta[1] = 0.0;
    // прямой ход прогонки
    for (int i = 1; i < N; i++) {
        alfa[i + 1] = b[i] / (c[i] - a[i] * alfa[i]);
        beta[i + 1] = (fi[i] + a[i] * beta[i]) / (c[i] - alfa[i] * a[i]);
    }
    // обратный ход прогонки
    y[N] = 1;
    for (int i = N - 1; i >= 0; i--) {
        y[i] = alfa[i + 1] * y[i + 1] + beta[i + 1];
    }
    delete[]a;
    delete[]b;
    delete[]c;
    delete[]alfa;
    delete[]beta;
}
void real_decision(double* u, int N) {
    double h = 1.0 / N;
    double* x;
    x = new double[N + 1];
    for (int i = 1; i < N + 1; i++) {
        x[i] = i * h;
    }
```

```
    double k1 = exp(-x0 * x0);
    double q1 = x0 * x0;
    double fi1 = sin(x0);
    double d1 = fi1 / q1; // частное решение

    double k2 = x0;
    double q2 = x0 * x0;
    double fi2 = sin(x0);
    double d2 = fi2 / q2;

    double lambda1 = sqrt(q1 / k1);
    double lambda2 = sqrt(q2 / k2);

    double A1 = exp(-lambda1 * x0) - exp(lambda1 * x0);
    double B1 = exp(-2 * lambda2 + lambda2 * x0) - exp(-lambda2 * x0);
    double D1 = d1 * (exp(lambda1 * x0) - 1) + exp(lambda2 * x0 - lambda2) * (1 - d2) + d2;

    double A2 = -k1 * (lambda1 * exp(lambda1 * x0) + lambda1 * exp(-lambda1 * x0));
    double B2 = k2 * (lambda2 * exp(-2 * lambda2 + lambda2 * x0) + lambda2 * exp(-lambda2
* x0));
    double D2 = k1 * d1 * lambda1 * exp(lambda1 * x0) + k2 * lambda2 * exp(-lambda2 +
lambda2 * x0) * (1 - d2);

    double znam= A1 * B2 - B1 * A2;

    double const2 = (D1 * B2 - B1 * D2) /znam;
    double const4 = (A1 * D2 - A2 * D1) /znam;
    double const1 = -d1 - const2;
    double const3 = exp(-lambda2) * (1 - d2 - const4 * exp(-lambda2));

    u[0] = 0;
    u[N] = 1;
    for (int i = 1; i < N; i++) {
        if (x[i] < x0)
            u[i] = const1 * exp(lambda1 * x[i]) + const2 * exp(-lambda1 * x[i]) + d1;
        else
            u[i] = const3 * exp(lambda2 * x[i]) + const4 * exp(-lambda2 * x[i]) + d2;
    }
    delete[] x;
}
double max_delta(double* u1, double* u2, int N) {
    double Max = 0.0;
    for (int i = 0; i < N + 1; i++) {
        if (Max < abs(u1[i] - u2[i])) {
            Max = abs(u1[i] - u2[i]);
        }
    }
    return Max;
```

```
}

int main() {
    int N, m;
    double delta, maximum;
    double* q, * k, * y, * u, * x, * fi;
    cout << "Start program! Enter N, please:" << endl;
    cin >> N;
    double h = 1.0 / N; // шаг
    cout << "Model or no ? Enter m,please: " << endl;
    cin >> m;
    if (m == 0) {
        cout << " model " << endl;
    }
    else {
        cout << "no model" << endl;
    }
    while (N < 10000) {
        if (m == 0) {
            u = new double[N + 1];
            real_decision(u, N);
            x = new double[N + 1];
            y = new double[N + 1];
            q = new double[N + 1];
            k = new double[N + 1];
            fi = new double[N + 1];
            progonka(x, y, k, q, fi, N,m);
            for (int i = 0; i < N + 1; i++) {
                cout <<"x="<< x[i]<< "  u[" << x[i] << "] =" << u[i] << "  y["<< x[i]<<"] =" << y[i] << "
delta = " << abs(y[i] - u[i]) << '\n';
            }
            delta = max_delta(u, y, N);
            if (max_delta(u, y, N) < 0.01) {
                cout << "max delta = " << max_delta(u, y, N) << " < 0.01 " << '\n';
                break;
            }
            else {
                cout << "max delta = " << max_delta(u, y, N) << " > 0.01 " << '\n';
                N = 2 * N;
                cout << "New N = " << N << '\n';
            }
            delete[] x;
            delete[] k;
            delete[] q;
            delete[] fi;
            delete[] y;
            delete[] u;
```

```
        }  else {
            maximum = 0;
            x = new double[N + 1];
            y = new double[N + 1];
            q = new double[N + 1];
            k = new double[N + 1];
            fi = new double[N + 1];

            progonka(x, y, k, q, fi, N,m);
            delete[] x;
            delete[] k;
            delete[] q;
            delete[] fi;

            x = new double[2*N + 1];
            q = new double[2*N + 1];
            k = new double[2*N + 1];
            fi = new double[2*N + 1];
            u = new double[2*N + 1];
            progonka(x, u, k, q, fi, 2*N,m);
            for (int i = 0; i < N + 1; i++) {
                cout <<" x = " << x[2*i] <<" y_N[" << x[2 * i] << "] = " << y[i] << " y_2N[" << x[2 * i]
<< "] = " << u[2 * i] << " delta = " << abs(u[2 * i] - y[i]) << '\n';
                if (maximum < abs(u[2 * i] - y[i]))
                    maximum = abs(u[2 * i] -y[i]);
            }
            if (maximum < 0.01) {
                cout << "max delta = " << maximum << " < 0.01 " << '\n';
                break;
            }
            else {
                cout << "max delta = " << maximum << " > 0.01 " << '\n';
                N = 2*N;
                cout << "New N = " << N << '\n';
            }

            delete[] x;
            delete[] k;
            delete[] q;
            delete[] fi;
            delete[] y;
            delete[] u;
        }
    }
    return 0;
}
```

Приложение

$$\begin{cases} \frac{d}{dx}\left(\bar{k}\frac{du}{dx}\right) - \bar{q}u = -\bar{f} \\ u(0) = 0,\ u(1) = 1 \end{cases}$$

$\bar{k} = k_1(x_0) = k_1,\ \bar{q} = q_1(x_0) = q_1,\ \bar{f} = f_1(x_0) = f_1, \quad 0 < x < x_0$

$\bar{k} = k_2(x_0) = k_2,\ \bar{q} = q_2(x_0) = q_2,\ \bar{f} = f_2(x_0) = f_2, \quad x_0 < x < 1$

Общее решение

$u = u_{общ} + u_{част}$

Находим общее решение однородного дифф. ур-ния второго порядка

подстановка $u = e^{\lambda x}$

$$\begin{cases} k_1\lambda_1^2 + q_1 = 0, & 0 < x < x_0 \\ k_2\lambda_2^2 + q_2 = 0, & x_0 < x < 1 \end{cases}$$

$\lambda_1 = \pm\sqrt{\frac{q_1}{k_1}},\ \lambda_2 = \pm\sqrt{\frac{q_2}{k_2}}$

$$\begin{cases} u_1 = C_1 e^{\lambda_1 x} + C_2 e^{-\lambda_1 x}, & 0 < x < x_0 \\ u_2 = C_3 e^{\lambda_2 x} + C_3 e^{-\lambda_2 x}, & x_0 < x < 1 \end{cases}$$

$$\begin{cases} u_{част_1} = \frac{f_1}{q_1}, & 0 < x < x_0 \\ u_{част_2} = \frac{f_2}{q_2}, & x_0 < x < 1 \end{cases}$$

Получим:

$$\begin{cases} u = C_1 e^{\lambda_1 x} + C_2 e^{-\lambda_1 x} + \frac{f_1}{q_1}, & 0 < x < x_0 \\ u = C_3 e^{\lambda_2 x} + C_4 e^{-\lambda_2 x} + \frac{f_2}{q_2}, & x_0 < x < 1 \end{cases}$$

краевые условия: $u(0) = 0,\ u(1) = 1$

условия сопряжения: $u\big|_{x = x_0 - 0} = u\big|_{x = x_0 + 0}$

$$k(x)\frac{du}{dx}\Big|_{x = x_0 - 0} = k(x)\frac{du}{dx}\Big|_{x = x_0 + 0}$$

Подставим

$$
\begin{cases}
C_1 + C_2 + \frac{f_1}{q_1} = 0 \\
C_3 e^{\lambda_2} + C_4 e^{-\lambda_2} + \frac{f_2}{q_2} = 1 \\
C_1 e^{\lambda_1 x_0} + C_2 e^{-\lambda_1 x_0} + \frac{f_1}{q_1} = C_3 e^{\lambda_2 x_0} + C_4 e^{-\lambda_2 x_0} + \frac{f_2}{q_2} \\
K_1 C_1 \lambda_1 e^{\lambda_1 x_0} - K_1 \lambda_1 C_2 e^{-\lambda_1 x_0} = K_2 C_3 \lambda_2 e^{\lambda_2 x_0} - K_2 C_4 \lambda_2 e^{-\lambda_2 x_0}
\end{cases}
$$

$$C_1 = -\frac{f_1}{q_1} - C_2$$

$$C_3 = \left(1 - \frac{f_2}{q_2} - C_4 e^{-\lambda_2}\right) e^{-\lambda_2}$$

$$\left(-\frac{f_1}{q_1} - C_2\right) e^{\lambda_1 x_0} + C_2 e^{-\lambda_1 x_0} + \frac{f_1}{q_1} = \left(1 - \frac{f_2}{q_2} - C_4 e^{-\lambda_2}\right) e^{-\lambda_2} e^{\lambda_2 x_0} + C_4 e^{-\lambda_2 x_0} + \frac{f_2}{q_2} \quad (*)$$

$$K_1\left(-\frac{f_1}{q_1} - C_2\right)\lambda_1 e^{\lambda_1 x_0} - K_1 \lambda_1 C_2 e^{-\lambda_1 x_0} = K_2\left(1 - \frac{f_2}{q_2} - C_4 e^{-\lambda_2}\right) e^{-\lambda_2} \lambda_2 e^{\lambda_2 x_0} - K_2 C_4 \lambda_2 e^{-\lambda_2 x_0} \quad (**)$$

(*):

$$-\frac{f_1}{q_1} e^{\lambda_1 x_0} - C_2 e^{\lambda_1 x_0} + C_2 e^{-\lambda_1 x_0} = e^{\lambda_2(x_0-1)} - \frac{f_2}{q_2} e^{\lambda_2(x_0-1)} - C_4 e^{\lambda_2(x_0-2)} + C_4 e^{-\lambda_1 x_0} + \frac{f_2}{q_2}$$

$$-\frac{f_1}{q_1} e^{\lambda_1 x_0} + C_2\left(e^{-\lambda_1 x_0} - e^{\lambda_1 x_0}\right) - e^{\lambda_2(x_0-1)} + \frac{f_2}{q_2} e^{\lambda_2(x_0-1)} + \underbrace{C_4 e^{\lambda_2(x_0-2)} - C_4 e^{-\lambda_2 x_0}}_{C_4\left(e^{\lambda_2(x_0-2)} - e^{-\lambda_2 x_0}\right)} - \frac{f_2}{q_2} = 0$$

$$\exists \quad e^{-\lambda_1 x_0} - e^{\lambda_1 x_0} = A_1$$

$$e^{\lambda_2(x_0-2)} - e^{-\lambda_2 x_0} = B_1$$

$$\frac{f_1}{q_1}\left(e^{\lambda_1 x_0} - 1\right) + e^{\lambda_2(x_0-1)} \cdot \left(1 - \frac{f_2}{q_2}\right) + \frac{f_2}{q_2} = D_1$$

Получим $C_2 A_1 + C_4 B_1 = D_1$

(**)

$$K_1\left(-\frac{f_1}{q_1} - C_2\right)\lambda_1 e^{\lambda_1 x_0} - K_1 C_2 \lambda_1 e^{-\lambda_1 x_0} - K_2\left(1 - \frac{f_2}{q_2} - C_4 e^{-\lambda_2}\right) e^{-\lambda_2} \lambda_2 e^{\lambda_2 x_0} + K_2 C_4 \lambda_2 e^{-\lambda_2 x_0}$$

$$-\frac{f_1}{q_1} K_1 \lambda_1 e^{\lambda_1 x_0} - C_2 K_1 \lambda_1 e^{\lambda_1 x_0} - K_1 C_2 \lambda_1 e^{-\lambda_1 x_0} - K_2 e^{-\lambda_2} \lambda_2 e^{\lambda_2 x_0} + \frac{f_2}{q_2} K_2 e^{-\lambda_2} \lambda_2 e^{\lambda_2 x_0} + C_4 e^{-2\lambda_2 + \lambda_2 x_0} K_2 \cdot \lambda_2 + K_2 C_4 \lambda_2 e^{-\lambda_2 x_0}$$

$$C_2 \cdot \underbrace{-K_1 \cdot \left(\lambda_1 e^{\lambda_1 x_0} + \lambda_1 e^{-\lambda_1 x_0}\right)}_{A_2} \qquad C_4 \cdot \underbrace{K_2\left(\lambda_2 e^{\lambda_2(x_0-2)} + \lambda_2 e^{-\lambda_2 x_0}\right)}_{B_2}$$

$$-\frac{f_1}{q_1} K_1 \lambda_1 e^{\lambda_1 x_0} + C_2 A_2 - K_2 e^{-\lambda_2} \lambda_2 e^{\lambda_2 x_0} + \frac{f_2}{q_2} K_2 e^{-\lambda_2} e^{\lambda_2 x_0} \lambda_2 + C_4 B_2 = 0$$

$$-\frac{f_1}{q_1} K_1 \lambda_1 e^{\lambda_1 x_0} - K_2 \lambda_2 e^{x_0(1-\lambda_2)} + \frac{f_2}{q_2} K_2 \lambda_2 e^{\lambda_2(x_0-1)} + C_2 A_2 + C_4 B_2$$

$$C_2 A_2 + C_4 B_2 = \underbrace{\frac{f_1}{q_1} K_1 \lambda_1 e^{\lambda_1 x_0} + K_2 e^{\lambda_2(x_0-1)} \lambda_2 \cdot \left(1 - \frac{f_2}{q_2}\right)}_{D_2}$$

$$A_2 = -k_1(\lambda_1 e^{\lambda_1 x_0} + \lambda_1 e^{-\lambda_1 x_0})$$

$$B_2 = k_2(\lambda_2 e^{\lambda_2(x_0-2)} + \lambda_2 e^{-\lambda_2 x_0})$$

$$D_2 = \frac{f_1}{q_1} k_1 \lambda_1 e^{\lambda_1 x_0} + k_2 e^{\lambda_2(x_0-1)} \cdot \lambda_2 \cdot (1 - \frac{f_2}{q_2})$$

Получим $C_2 A_2 + C_4 B_2 = D_2$

Получим систему

$$\begin{cases} C_2 A_1 + C_4 B_1 = D_1 \\ C_2 A_2 + C_4 B_2 = D_2 \end{cases}$$

$$C_4 = \frac{D_2 A_1 - D_1 A_2}{A_1 B_2 - B_1 A_2}$$

$$C_2 = \frac{D_1 B_2 - D_2 B_1}{A_1 B_2 - B_1 A_2}$$

Вспомним, что:

$$C_1 = -\frac{f_1}{q_1} - C_2$$

$$C_3 = (1 - \frac{f_2}{q_2} - C_4 e^{-\lambda_2}) e^{-\lambda_2}$$