Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № *4*

Дисциплина*:* Моделирование

Тема: Программно-алгоритмическая реализация моделей на основе дифференциальных уравнений в частных производных с краевыми условиями II и III рода.

Студент Лаврова А. А.

Группа ИУ7-65Б

Оценка (баллы) ________________

Преподаватель Градов В.М.

Цель работы: получение навыков разработки алгоритмов решения смешанной краевой задачи при реализации моделей, построенных на квазилинейном уравнении параболического вида.

Исходные данные:

1. Уравнение для функции T(x):
$$c(T)\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k(T)\frac{\partial T}{\partial x}\right) - \frac{2}{R}\alpha(x)T + \frac{2T_0}{R}\alpha(x)$$

2. Краевые условия:
$$\begin{cases} t = 0, \quad T(x,0) = T_0 \\ x = 0, \quad -k\big(T(0)\big)\dfrac{\partial T}{\partial x} = F_0 \\ x = l, \quad -k\big(T(l)\big)\dfrac{\partial T}{\partial x} = \alpha_N(T(l) - T_0) \end{cases}$$

3. В обозначения уравнения (14.1):
$$p(x) = \frac{2}{R}\alpha(x), \qquad f(u) \equiv f(x) = \frac{2T_0}{R}\alpha(x)$$

4. Функция $\alpha(x)$ задана своими константами:
$$\alpha(x) = \frac{c}{x-d}$$
где
$$\alpha(0) = \alpha_0, \qquad \alpha(l) = \alpha\_N$$
$$c = -\alpha_0 d, \quad d = \frac{\alpha_0 l}{a_N - \alpha_0}$$

5. Разностная схема с разностным краевым условием при x=0 получена в лекции №14 и может быть использована в данной работе. Получим интегро-интерполяционным методом разностный аналог краевого условия при $x = l$ так же, как это сделано при $x = 0$ в лекции №14.

   Разностная схема:

$$\widehat{A_n}\,\widehat{y_{n-1}} - \widehat{B_n}\,\widehat{y_n} + \widehat{D_n}\,\widehat{y_{n+1}} = -\widehat{F_n}\ , 1 \le n \le N-1$$
$$\widehat{A_n} = \widehat{X_{n-\frac{1}{2}}}\,\frac{\tau}{h}$$
$$\widehat{B_n} = \widehat{A_n} + \widehat{D_n} + \widehat{c_n}\,h + p_n h\tau$$
$$\widehat{D_n} = \widehat{X_{n+\frac{1}{2}}}\,\frac{\tau}{h}$$
$$\widehat{F_n} = f_n\,\tau\,h + \widehat{c_n}\,y_n h$$

Разностные аналоги краевых условий при x = 0:

$$\left(\frac{h}{8}\widehat{c_{\frac{1}{2}}} + \frac{h}{4}\,\widehat{c_0} + \widehat{X_{\frac{1}{2}}}\frac{\tau}{h} + \frac{\tau h}{8}\,p_{\frac{1}{2}} + \frac{\tau h}{4}p_0\right)\widehat{y_0}$$
$$+\left(\frac{h}{8}\,\widehat{c_{\frac{1}{2}}} - \widehat{X_{\frac{1}{2}}}\frac{\tau}{h} + \frac{\tau h}{8}p_{\frac{1}{2}}\right)\widehat{y_1}$$
$$= \frac{h}{8}\,\widehat{c_{\frac{1}{2}}}\,(y_0 + y_1) + \frac{h}{4}\,\widehat{c_0}\,y_0 + \hat{F}\,\tau + \frac{\tau h}{4}\,\left(\widehat{f_{\frac{1}{2}}} + \widehat{c_0}\right)$$

Проинтегрируем уравнение:

$$F = -k(u)\frac{\partial T}{\partial x}$$

для функции $T(x,t)$ на отрезке $[X_{N-\frac{1}{2}},\ X_N]$ с учетом того, что поток:

$$F_N = \alpha_N(\widehat{y_N} - T_0)$$
$$F_{N-\frac{1}{2}} = \widehat{X_{N-\frac{1}{2}}}\,\frac{\widehat{y_{N-1}} - \widehat{y_N}}{h}$$

Получаем:

$$\int_{X_{N-\frac{1}{2}}}^{X_N} dx \int_{t_m}^{t_{m+1}} c(t)\frac{\partial T}{\partial t}dt$$
$$= -\int_{t_m}^{t_{m+1}} dt \int_{X_{N-\frac{1}{2}}}^{X_N} \frac{\partial F}{\partial x}\,dx - \int_{X_{N-\frac{1}{2}}}^{X_N} p(x)dx \int_{t_m}^{t_{m+1}} Tdt$$
$$+ \int_{X_{N-\frac{1}{2}}}^{X_N} f(x)dx \int_{t_m}^{t_{m+1}} dx$$

Вычислим интегралы по t методом правым прямоугольников:

$$\int_{X_{N-\frac{1}{2}}}^{X_N} \hat{c}\,\left(\hat{T} - T\right)dx$$
$$= -\int_{t_m}^{t_{m+1}} \left(F_N - F_{N-\frac{1}{2}}\right)dt - \int_{X_{N-\frac{1}{2}}}^{X_N} p\,\hat{T}\,\tau\,dx$$
$$+ \int_{t_m}^{t_{m+1}} \hat{f}\,\tau\,dx$$

Вычислим интегралы по x методом трапеций:

$$\frac{h}{4}\left(\widehat{c_N}\left(\widehat{y_N}-y_N\right)-\widehat{c_{N-\frac{1}{2}}}\left(\frac{\widehat{y_N}+\widehat{y_{N-1}}}{2}-\frac{y_N+y_{N+1}}{2}\right)\right)$$
$$=-\tau\left(\alpha_N\left(\widehat{y_N}-T_0\right)-\widehat{X_N}\frac{\widehat{y_N}+\widehat{y_{N-1}}}{h}\right)$$
$$-\left(p_N\widehat{y_N}-p_{n-\frac{1}{2}}\frac{\widehat{y_N}+\widehat{y_{N-1}}}{2}\right)\frac{\tau h}{4}+\left(\widehat{f_N}-\widehat{f_{N-\frac{1}{2}}}\right)\frac{\tau h}{4}$$

$$\left(\frac{h}{4}\widehat{c_N}+\frac{h}{8}\widehat{c_{N-\frac{1}{2}}}+\tau\,\alpha_N+\frac{\tau}{h}\widehat{X_{N-\frac{1}{2}}}+\frac{h}{4}\tau\,p_N+\frac{h}{8}\tau\,p_{N-\frac{1}{2}}\right)\widehat{y_N}$$
$$+\left(\frac{h}{8}\widehat{C_{N-\frac{1}{2}}}-\frac{\tau}{h}\widehat{X_{N-\frac{1}{2}}}+\frac{h}{8}\tau\,p_{N-\frac{1}{2}}\right)\widehat{y_{N-1}}$$
$$=\alpha_N\,\tau\,T_0+\frac{h}{4}\widehat{C_N}y_N+\frac{h}{8}\widehat{c_{N-\frac{1}{2}}}\left(y_N+y_{N-1}\right)+\frac{h}{4}\tau\left(\widehat{f_N}\right.$$
$$\left.+\widehat{f_{N-\frac{1}{2}}}\right)$$

Из разностного аналога краевых условий при x = 0 и из приведенной выше формулы получим коэффициенты $K_0, M_0, P_0, K_N, M_N, P_N$:

$$\begin{cases} \widehat{K_0}\,\widehat{y_0}+\widehat{M_0}\,\widehat{y_1}=\widehat{P_0} \\ \widehat{A_n}\,\widehat{y_{n-1}}-\widehat{B_n}\,\widehat{y_n}+\widehat{D_n}\,\widehat{y_{n+1}}=-\widehat{F_n}\ , \quad 1\le n\le N-1 \\ \widehat{K_N}\,\widehat{y_N}+\widehat{M_{N-1}}\,\widehat{y_{N-1}}=\widehat{P_N} \end{cases}$$

Данная система решается методом итераций. Обозначим текущую итерацию как s, а предыдущую – как s-1:

$$A_n^{s-1}\,y_{n+1}^s-B_n^{s-1}\,y_n^s+D_n^{s-1}\,y_{n-1}^s=-F_n^{s-1}$$

6. Значения параметров для отладки:

$k(T)=a_1(b_1+c_1\,T^{m_1}),\qquad \frac{\text{Вт}}{\text{см}*\text{К}}$

$c(T)=a_2+b_2\,T^{m_2}-\frac{c_2}{T^2}\ ,\qquad \frac{\text{Дж}}{\text{см}^3*\text{К}}$

$a_1=0.0134,\qquad b_1=1,\qquad c_1=4.35*10^{-4},\qquad m_1=1$

$a_2=2.049,\qquad b_2=0.563*10^{-3},\qquad c_2=0.528*10^{5},\qquad m_2=1$

$$\alpha(x) = \frac{c}{x-d}$$

$$\alpha_0 = 0.05\ \frac{\text{Вт}}{\text{см}^2 * \text{К}}$$

$$\alpha_N = 0.01\ \frac{\text{Вт}}{\text{см}^2 * \text{К}}$$

$$l = 10\ \text{см}$$

$$T_0 = 300\ \text{К}$$

$$R = 0.5\ \text{см}$$

$$F(t) = 50\ \frac{\text{Вт}}{\text{см}^2}$$

Физическая составляющая задачи:

Постановки задач в данной лабораторной работе и работе №3 во многом совпадают. Отличия заключаются в следующем:

1. Сформулированная в данной работе математическая модель описывает нестационарное температурное поле T(x,t), зависящее от координаты x и меняющееся во времени.
2. Свойства материала стержня привязаны к температуре, т.е. теплоемкость и коэффициент теплопроводности c(T), k(T) зависят от T, тогда как в работе №3 k(x) зависит от координаты, а c = 0.
3. При x = 0 цилиндр нагружается тепловым потоком F(t), в общем случае зависящим от времени, а в работе №3 поток был постоянный.

Если в настоящей работе задать поток постоянным, т.е. $F(t)$=const, то будет происходить формирование температурного поля от начальной температуры T_0 до некоторого установившегося (стационарного) распределения $T(x,t)$. Это поле в дальнейшем с течением времени меняться не будет и должно совпасть с температурным распределением $T(x)$, получаемым в лаб. работе №3, если все параметры задач совпадают, в частности, вместо $k(T)$ надо использовать $k(x)$ из лаб. работы №3. Это полезный факт для тестирования программы. Если после разогрева стержня положить поток $F(t)$=0, то будет происходить остывание, пока температура не выровняется по всей длине и не станет равной T_0. При произвольной зависимости потока $F(t)$ от времени температурное поле будет как-то сложным образом отслеживать поток.

Практическая часть:

Листинг программы:

```
import matplotlib.pyplot as plt
import numpy as np
from math import fabs

a1 = 0.0134
b1 = 1
c1 = 4.35e-4
m1 = 1
a2 = 2.049
b2 = 0.563e-3
c2 = 0.528e5
m2 = 1
alpha0 = 0.05
alphaN = 0.01
l = 10
T0 = 300
R = 0.5
F0 = 50

h = 1e-3
t = 1


def k(T):
    return a1 * (b1 + c1 * T ** m1)


def c(T):
    return a2 + b2 * T ** m2 - (c2 / T ** 2)


def alpha(x):
    d = (alphaN * l) / (alphaN - alpha0)
    c = - alpha0 * d
    return c / (x - d)


def p(x):
    return 2 * alpha(x) / R


def f(x):
    return 2 * alpha(x) * T0 / R


def func_plus_half(x, step, func):
    return (func(x) + func(x + step)) / 2


def func_minus_half(x, step, func):
    return (func(x) + func(x - step)) / 2


def A(T):
    return t / h * func_minus_half(T, t, k)


def D(T):
```

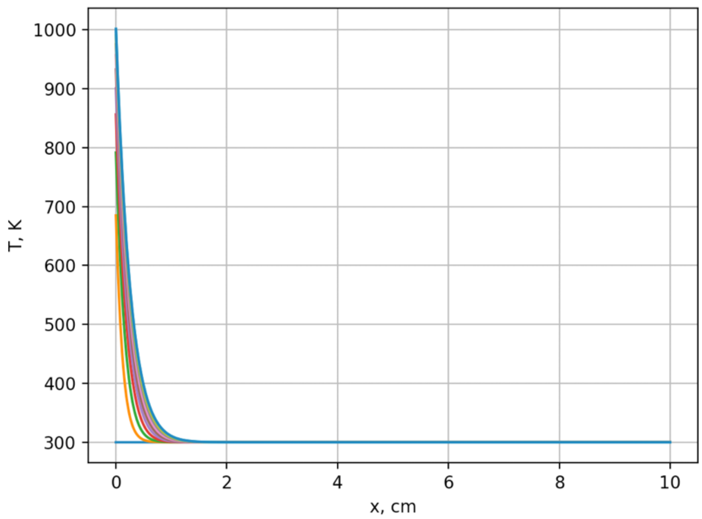
```
    return t / h * func_plus_half(T, t, k)


def B(x, T):
    return A(T) + D(T) + c(T) * h + p(x) * h * t


def F(x, T):
    return f(x) * h * t + c(T) * T * h


def left_boundary_condition(T_prev):
    K0 = h / 8 * func_plus_half(T_prev[0], t, c) + h / 4 * c(T_prev[0]) + \
         func_plus_half(T_prev[0], t, k) * t / h + \
         t * h / 8 * p(h / 2) + t * h / 4 * p(0)

    M0 = h / 8 * func_plus_half(T_prev[0], t, c) - \
         func_plus_half(T_prev[0], t, k) * t / h + \
         t * h * p(h / 2) / 8

    P0 = h / 8 * func_plus_half(T_prev[0], t, c) * (T_prev[0] + T_prev[1]) +
\
         h / 4 * c(T_prev[0]) * T_prev[0] + \
         F0 * t + t * h / 8 * (3 * f(0) + f(h))

    return K0, M0, P0


def right_boundary_condition(T_prev):
    KN = h / 8 * func_minus_half(T_prev[-1], t, c) + h / 4 * c(T_prev[-1]) +
\
         func_minus_half(T_prev[-1], t, k) * t / h + t * alphaN + \
         t * h / 8 * p(l - h / 2) + t * h / 4 * p(l)

    MN = h / 8 * func_minus_half(T_prev[-1], t, c) - \
         func_minus_half(T_prev[-1], t, k) * t / h + \
         t * h * p(l - h / 2) / 8

    PN = h / 8 * func_minus_half(T_prev[-1], t, c) * (T_prev[-1] + T_prev[-
2]) + \
         h / 4 * c(T_prev[-1]) * T_prev[-1] + t * alphaN * T0 + \
         t * h / 4 * (f(l) + f(l - h / 2))

    return KN, MN, PN


def run(T_prev, K0, M0, P0, KN, MN, PN):
    eps = [0, -M0 / K0]
    eta = [0, P0 / K0]

    x = h
    n = 1
    while (x + h < l):
        eps.append(D(T_prev[n]) / (B(x, T_prev[n]) - A(T_prev[n]) * eps[n]))
        eta.append((F(x, T_prev[n]) + A(T_prev[n]) * eta[n]) / (B(x,
T_prev[n]) - A(T_prev[n]) * eps[n]))
        n += 1
        x += h

    y = [0] * (n + 1)
    y[n] = (PN - MN * eta[n]) / (KN + MN * eps[n])

    for i in range(n - 1, -1, -1):
        y[i] = eps[i + 1] * y[i + 1] + eta[i + 1]
```

```python
    return y


def main():
    step1 = int(l / h)
    T = [T0] * (step1 + 1)
    T_new = [0] * (step1 + 1)
    ti = 0
    res = []
    res.append(T)

    while True:
        prev = T
        while True:
            K0, M0, P0 = left_boundary_condition(prev)
            KN, MN, PN = right_boundary_condition(prev)
            T_new = run(prev, K0, M0, P0, KN, MN, PN)

            max = fabs((T[0] - T_new[0]) / T_new[0])
            for step2, j in zip(T, T_new):
                d = fabs(step2 - j) / j
                if d > max:
                    max = d
            if max < 1:
                break

            prev = T_new
        res.append(T_new)
        ti += t

        check_eps = 0
        for i, j in zip(T, T_new):
            if fabs((i - j) / j) > 1e-2:
                check_eps = 1
        if check_eps == 0:
            break
        T = T_new

    x = [i for i in np.arange(0, l, h)]
    te = [i for i in range(0, ti, t)]

    step1 = 0
    for i in res:
        if (step1 % 2 == 0):
            plt.plot(x, i[:-1])
        step1 += 1
    plt.plot(x, res[-1][:-1])
    plt.xlabel("x, cm")
    plt.ylabel("T, K")
    plt.grid()
    plt.show()

    step2 = 0
    while (step2 < l / 3):
        point = [j[int(step2 / h)] for j in res]
        plt.plot(te, point[:-1])
        step2 += 0.1
    plt.xlabel("t, sec")
    plt.ylabel("T, K")
    plt.grid()
    plt.show()
```

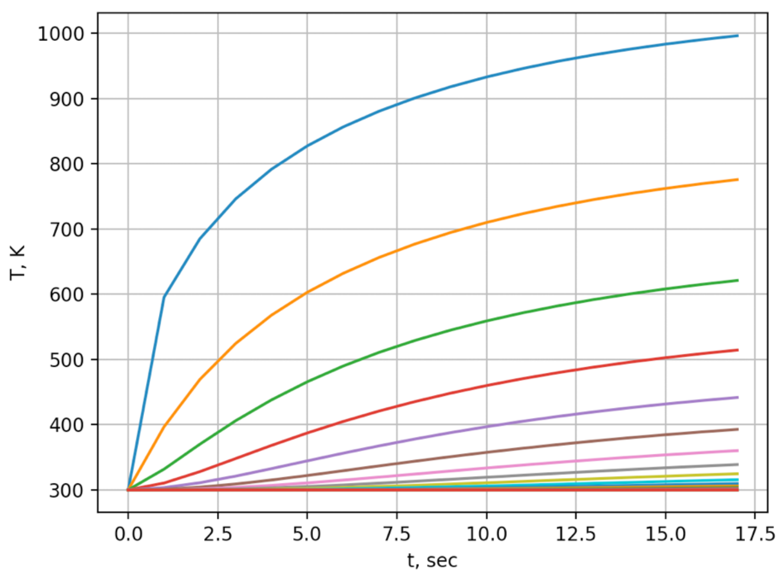
```python
if __name__ == "__main__":
    main()
```

Тестирование:

1. График зависимости температуры $T(x,t_{\mathrm{m}})$ от координаты x при нескольких фиксированных значениях времени t_{m} при заданных выше параметрах.



2. График зависимости $T(x_{\mathrm{n}}, t)$ при нескольких фиксированных значениях координаты x_{n}.
(верхний график соответствует случаю x = 0, нижний – x=l)



Контрольные вопросы:

1. *Приведите результаты тестирования программы (графики, общие соображения, качественный анализ). Учесть опыт выполнения лабораторной работы №3.*

Был рассмотрен случай, когда для получения аналогичного графика из лабораторной работы №3 значение τ приравнивается к 1, обнулено $c(u)$, а также зависимость коэффициента теплопроводности k от температуры изменена на зависимость от координаты x на стержне.

График из лабораторной работы №3:

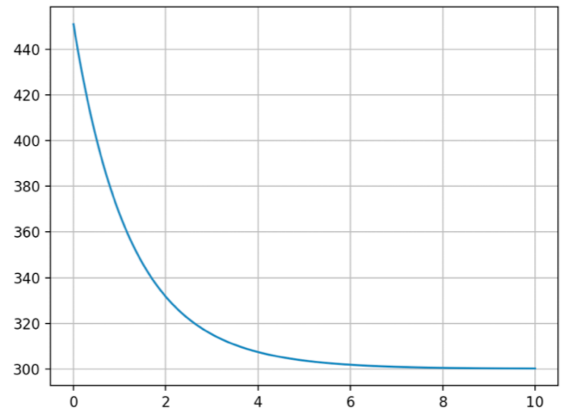
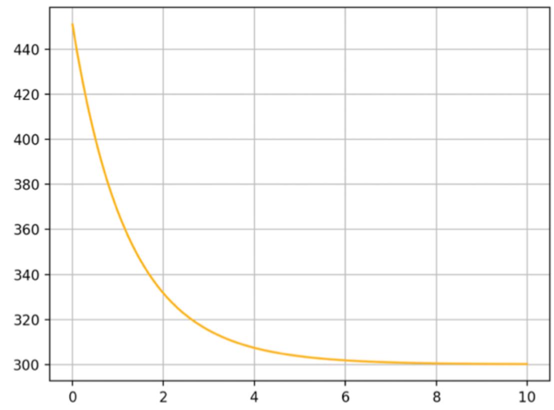
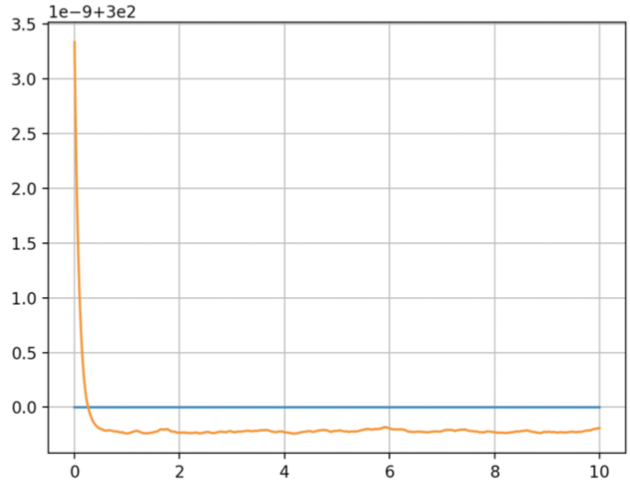


График из лабораторной работы №4:



В случае обнуления потока $F_0(T)$ на выходе будет график температуры, которая получилась благодаря температуре окружающей среды.

График из лабораторной работы №4 при нулевом потоке:



2. *Выполните линеаризацию уравнения по Ньютону, полагая для простоты, что все коэффициенты зависят только от одной переменной $\widehat{y_n}$*

   Необходимо линеаризовать систему уравнений, используя метод Ньютона:

$$\begin{cases} \widehat{K_0}\widehat{y_0} + \widehat{M_0}\widehat{y_1} = \widehat{P_0} \\ \widehat{A_n}\widehat{y_{n-1}} - \widehat{B_n}\widehat{y_n} + \widehat{D_n}\widehat{y_{n+1}} = -\widehat{F_n}\,, \quad 1 \le n \le N-1 \\ \widehat{K_N}\widehat{y_n} + \widehat{M_{N-1}}\widehat{y_{N-1}} = \widehat{P_N} \end{cases}$$

   Зная, что коэффициенты зависят от $\widehat{y_n}$, получим:

$$\left(\widehat{A_n}\widehat{y_{n-1}} - \widehat{B_n}\widehat{y_n} + \widehat{D_n}\widehat{y_{n+1}} + \widehat{F_n}\right)\Big|_{s-1} + \widehat{A_n}^{(s-1)}\Delta\,\widehat{y_{n-1}}^{(s)} + \left(\frac{\partial\widehat{A_n}}{\partial\widehat{y_n}}\widehat{y_{n-1}} - \frac{\partial\widehat{B_n}}{\partial\widehat{y_n}}\widehat{y_n} - \widehat{B_n} + \frac{\partial\widehat{D_n}}{\partial\widehat{y_n}}\widehat{y_{n+1}} + \frac{\partial\widehat{F_n}}{\partial\widehat{y_n}}\right)\Bigg|_{(s-1)} * \Delta\widehat{y_n}^{(s)} + \widehat{D_n}^{(s-1)}\Delta\widehat{y_{n+1}}^{(s)} = 0$$

Канонический вид:

$$A_n\Delta\,\widehat{y_{n-1}}^{(s)} - B_n\Delta\widehat{y_n}^{(s)} + D_n\,\Delta\widehat{y_{n+1}}^{(s)} = -F_n\,,$$

где:

$$A_n = \widehat{A_n}^{(s-1)}$$

$$B_n = \left(-\frac{\partial \widehat{A_n}}{\partial \widehat{y_n}}\widehat{y_{n-1}} + \frac{\partial \widehat{B_n}}{\partial \widehat{y_n}}\widehat{y_n} + \widehat{B_n} - \frac{\partial \widehat{D_n}}{\partial \widehat{y_n}}\widehat{y_{n+1}} - \frac{\partial \widehat{F_n}}{\partial \widehat{y_n}}\right)\Bigg|_{(s-1)}$$

$$D_n = \widehat{D_n}^{(s-1)}$$

$$F_n = \left(\widehat{A_n}\widehat{y_{n-1}} - \widehat{B_n}\widehat{y_n} + \widehat{D_n}\widehat{y_{n+1}} + \widehat{F_n}\right)\Big|_{s-1}$$

Краевые условия для y_0 и y_n:

$$\widehat{K_0} * \widehat{y_0} + \widehat{M_0} * \widehat{y_1} = \widehat{P_0}$$

$$\widehat{K_n} * \widehat{y_n} + \widehat{M_{N-1}} * \widehat{y_{N-1}} = \widehat{P_N}$$

Получаем:

$$(\widehat{K_0} * \widehat{y_0} + \widehat{M_0} * \widehat{y_1} - \widehat{P_0})\big|_{(s-1)} + \widehat{K_0}^{(s-1)} * \Delta\widehat{y_0}^{(s)} + \widehat{M_0}^{(s-1)} * \Delta\,\widehat{y_1}^{(s)} = 0$$

$$(\widehat{K_N} * \widehat{y_N} + \widehat{M_{N-1}} * \widehat{y_{N-1}} - \widehat{P_N})\big|_{(s-1)} + \widehat{K_N}^{(s-1)} * \Delta\widehat{y_N}^{(s)} + \widehat{M_{N-1}}^{(s-1)} * \Delta\,\widehat{y_{N-1}}^{(s)} = 0$$

Каноничный вид:

$$K_0 * \Delta\widehat{y_0}^{(s)} + M_0 * \Delta\widehat{y_1}^{(s)} = P_0$$

$$K_N * \Delta\widehat{y_N}^{(s)} + M_{N-1} * \Delta\widehat{y_{N-1}}^{(s)} = P_N$$

Получаем следующие коэффициенты:

$$K_0 = \widehat{K_0}^{(s-1)}$$

$$M_0 = \widehat{M_0}^{(s-1)}$$

$$P_0 = (\widehat{K_0} * \widehat{y_0} + \widehat{M_0} * \widehat{y_1} - \widehat{P_0})\big|_{(s-1)}$$

$$K_N = \widehat{K_N}^{(s-1)}$$

$$M_N = \widehat{M_{N-1}}^{(s-1)}$$

$$P_N = (\widehat{K_N} * \widehat{y_N} + \widehat{M_{N-1}} * \widehat{y_{N-1}} - \widehat{P_N})\big|_{(s-1)}$$

После этого получим систему:

$$\begin{cases} K_N * \ \Delta\widehat{y_N}^{(s)} + M_{N-1} * \Delta\widehat{y_{N-1}}^{(s)} = P_N \\ K_0 * \ \Delta\widehat{y_0}^{(s)} + M_0 * \Delta\widehat{y_1}^{(s)} = P_0 \\ A_n * \ \Delta\widehat{y_{n-1}}^{(s)} - \ B_n * \ \Delta\widehat{y_n}^{(s)} + \ D_n * \ \Delta\widehat{y_{n+1}}^{(s)} = \ -F_N \end{cases}$$

Для решения той системы необходимо найти все $\Delta\widehat{y_n}^{(s)}$. Зная приближение (s-1), можно найти приближение (s). Найдём значение искомой функции в узлах:

$$\widehat{y_n}^{(s)} = \ \widehat{y_n}^{(s-1)} + \ \widehat{y_n}^{(s)}$$

Итерационный процесс завершится при достижении условия:

$$\max\left|\frac{\Delta\widehat{y_n}^{(s)}}{\widehat{y_n}^{(s)}}\right| \leq \ \varepsilon, \qquad n = \overline{1, N}$$