Московский авиационный институт
(национальный исследовательский университет)
Кафедра 806
“Вычислительная математика и программирование”

КУРСОВАЯ РАБОТА
по дисциплине
“Численные методы”
Вариант №23

Студент группы М8О-308Б-18
_______________ Трофимов М.А.
(дата, подпись)

Преподаватель
_______________ Черкасов М.А.
(дата, подпись)

Москва, 2021

Условие:

```
308/23  Решить начально-краевую задачу для ДифУрЧаП параболического типа.
        Использовать схему: не явную.
        Осуществить реализацию варианта аппроксимации граничных условий,
        содержащих производные:
         - двухточечная с первым порядком точности


   y'  = 2.4* y''   +  6.30* y  + (x + 3 )/(t +   5 )  ;
    t          xx

    4 <= x <=   8 ; hx= 0.80000; ht=  0.02500; T_end=  0.10000

  5.00* Y'x (t,  4 )  +  4.00* Y (t,  4 ) =       78.83185307 + t / ( -4 )

  4.00* Y'x (t,  8 )  =       50.26548246 + t*t/( -3 ) -   0.800 *t  ;

  y(0,x)=   4 +   8 * sin(    1.57079633 * x )
```

Теория:

14. Решение уравнения теплопроводности. Метод сеток

Рассмотрим решение дифференциального уравнения в частных производных параболического типа с начально-краевыми условиями:

$$\frac{\partial y}{\partial t}=\alpha_1\cdot\frac{\partial^2 y}{\partial x^2}+\alpha_2\frac{\partial y}{\partial x}+\alpha_3 y+f(t,x),\quad a\le x\le b,\quad 0\le\ t\le T,\ \alpha_1>0 \qquad (14.1)$$

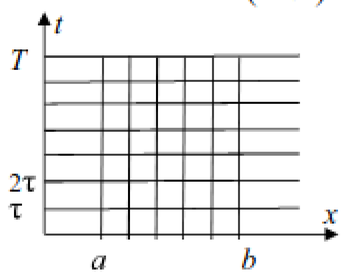
для $x=a$ $\qquad \varphi_1\frac{\partial y}{\partial x}+\varphi_2 y=f_1(t)$, $\qquad$ (14.2)

для $x=b$ $\qquad \varphi_4\frac{\partial y}{\partial x}+\varphi_5 y=f_2(t)$, $\qquad$ (14.3)

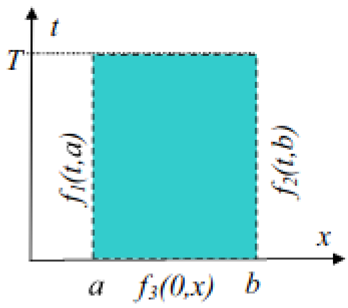
для $t=0$ $\qquad y(0,x) \quad =f_3(x)$. $\qquad$ (14.4)

(14.4)

Накроем область сеткой с шагом по x равным h и с шагом по t равным τ. Тогда $x_0=a$, $x_1=a+h$, …, $x_N=b$, $N=(b-a)/N$, $t_0=0$, $t_1=\tau$, $t_2=2\cdot\tau$, $t_M=T=M\cdot\tau$, $M=T/\tau$.



Геометрически область представляет собой «стакан», с трёх сторон которого заданы начальные условия (14.4), слева и справа заданы краевые условия (14.2) и (14.3), а на верхней кромке (при $t=T$) значения функции $y(t,x)$ не известны. Их вычисление и является целью рассматриваемых алгоритмов. Коэффициенты α_i уравнения и φ_i в краевых условиях представляют собой константы (могут быть и равны 0).



Тогда $y(t_k, x_i)=y_i^k$. Назовём её сеточной функцией. Рассмотрим несколько способов решения этой задачи.

14.2. Неявная схема

В (14.1) производные будем вычислять в узле $(x_i\,;\,t_{k+1})$. Заменим производные на конечно-разностные соотношения

$$\frac{\partial y}{\partial t}=\frac{y_i^{k+1}-y_i^k}{\tau}+O(\tau^1)\,;\quad \frac{\partial^2 y}{\partial x^2}=\frac{y_{i+1}^{k+1}-2y_i^{k+1}+y_{i-1}^{k+1}}{h^2}+O(h^2)\,;$$

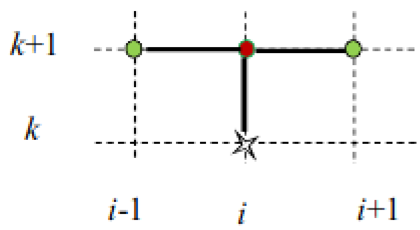
$$\frac{\partial y}{\partial x}=\frac{y_{i+1}^{k+1}-y_{i-1}^{k+1}}{2h}+O(h^2)\,. \qquad (14.11)$$

Подставим эти соотношения в (14.1) и получим для каждой внутренней точки $(x_i;\,t_{k+1})$:

$$(2\alpha_1-h\cdot\alpha_2)\frac{\tau}{2h^2}y_{i-1}^{k+1}+\left(\alpha_3\tau-\frac{2\alpha_1\tau}{h^2}-1\right)y_i^{k+1}+(2\alpha_1+h\cdot\alpha_2)\frac{\tau}{2h^2}y_{i+1}^{k+1}=$$

$$=-y_i^k-\tau\cdot f(t_{k+1};x_i)+O(\tau+h^2)\ ,\ i=1,2,3,\dots,N\text{-}1. \qquad (14.12)$$

Это уравнение содержит три неизвестных для i=1,3,…,n-1. Решение (14.1) по такой схеме носит название «неявная схема» и имеет графическое изображение:



Осталось дополнить систему (14.12) первым (для i=0, т.е. x=a) и последним (для i=N, т.е. x=b) уравнениями. Можно использовать формулы (14.7), но это ухудшит точность решения:

Для x=a первое уравнение: $(h\varphi_2-\varphi_1)y_0^1+\varphi_1 y_1^1=f_1(t_1)*h\,.$

Для x=b последнее уравнение: $-\varphi_4 y_{N-1}^1+(h\varphi_5+\varphi_4)y_N^1=f_2(t_1)*h\,.$ (14.13)

Используя (14.8), (14.9), (14.10), мы получим второй порядок точности по h, но эта трёхточечная схема сделает матрицу системы не трёхдиагональной:

Для x=a : $(2\,h\,\varphi_2-3\,\varphi_1)\,y_0+4\,\varphi_1\,y_1\ -\ \varphi_1\,y_2\ =2h\,f_1(t).$

Для x=b : $\varphi_4\,y_{N-2}-4\varphi_4\,y_{N-1}+(\,2h\varphi_5+3\varphi_4)\,y_{N-2}\ =2hf_1(t).$ (14.14)

Исправить её не трёхдиагональность можно арифметическими операциями со строками: первой со второй и последней с предпоследней. После такого исправления матрица системы станет трёхдиагональной, но вероятно потеряет свойство диагонального преобладания, что для матриц больших порядков может привести к потерям точности при применении метода прогонки.

Для сохранения трёхдиагональности матрицы системы и второго порядка точности вычислений относительно h, разложим y_1^{k+1} в ряд Тейлора в окрестности точки $(t_{k+1}; x_0)$:

$$y_1^{k+1} = y_0^{k+1} + \frac{\partial y}{\partial x} \cdot h + \frac{\partial^2 y}{\partial x^2} \cdot \frac{h^2}{2} + O(h^2). \qquad (14.15)$$

Сюда вместо y''_{xx} подставим его выражение из (14.1) и из полученного соотношения выразим y'_x в точке $(t_{k+1};x_0)$:

$$\left.\frac{\partial y}{\partial x}\right|_{i=0}^{k+1} = \frac{2\alpha_1}{h(2\alpha_1 - \alpha_2 h)}(y_1^{k+1} - y_0^{k+1}) - \frac{h}{2\alpha_1 - \alpha_2 h} \cdot \left.\frac{\partial y}{\partial t}\right|_0^{k+1} + \frac{\alpha_3 h}{2\alpha_1 - \alpha_2 h} y_0^{k+1} +$$

$$+ \frac{h}{2\alpha_1 - \alpha_2 h} f(t_{k+1}, x_0) + O(h^2).$$

Учтём, что $\left.\frac{\partial y}{\partial t}\right|_0^{k+1} = \frac{(y_0^{k+1} - y_0^k)}{\tau} + O(\tau)$ и получим первое уравнение (для $x=x_0$) будущей трёхдиагональной системы в случае, когда $\varphi_1 \neq 0$:

$$\left(\frac{2\alpha_1}{h} + \frac{h}{\tau} - \alpha_3 h - \frac{\varphi_2}{\varphi_1}(2\alpha_1 - \alpha_2 h)\right) y_0^{k+1} - \frac{2\alpha_1}{h} y_1^{k+1} =$$

$$= \frac{h}{\tau} y_0^k + h \cdot f(t_{k+1}, x_0) - \frac{2\alpha_1 - \alpha_2 h}{\varphi_1} f_1(t_{k+1}). \qquad (14.16)$$

Если $\varphi_1=0$, то первое уравнение будет выглядеть: $y_0^{k+1} = f_1(t_{k+1}) / \varphi_2$.

Аналогично последнее уравнение (для $x=x_N$) , будет для $\varphi_4 = 0$ выглядеть: $y_N^{k+1} = f_2(t_{k+1}) / \varphi_5$, а если $\varphi_4 \neq 0$ так:

$$-\frac{2\alpha_1}{h} y_{N-1}^{k+1} + \left(\frac{2\alpha_1}{h} + \frac{h}{\tau} - \alpha_3 h + \frac{\varphi_5}{\varphi_4}(2\alpha_1 + \alpha_2 h)\right) y_N^{k+1} =$$

$$= \frac{h}{\tau} y_N^k + h \cdot f(t_{k+1}, x_N) + \frac{2\alpha_1 + \alpha_2 h}{\varphi_4} f_2(t_{k+1}). \qquad (14.17)$$

Остальные N-1 уравнений для внутренних точек (для i=1,2,…,N-1) записываются по формуле (14.12).

При программировании неявной схемы надо учесть, что на каждом новом слое приходится решать систему линейных алгебраических уравнений с трёхдиагональной матрицей. Недостаток неявной схемы в необходимости решения трёхдиагональной СЛАУ (например, методом прогонки). Это несколько усложняет программирование, увеличивает количество арифметических операций (а значит ухудшение точности). Но решение получается устойчивым по сравнению с явной схемой. Поэтому вычисления возможно проводить с большим шагом по t, а это, даже с учётом метода прогонки, существенно уменьшает общее время вычисления до $t=T$, а значит может уменьшаться общее количество арифметических операций.

Подробное решение для первого шага:

$$\frac{\partial y}{\partial t} = 2.4 \cdot \frac{\partial^2 y}{\partial x^2} + 0 \cdot \frac{\partial y}{\partial x} + 6.3 \cdot y + \frac{x+3}{t+5}$$

$$x \in [4, 8],\ h = 0.8,\ t \in [0, 0.1],\ \tau = 0.025$$

$$5 \cdot \frac{\partial y}{\partial x}(t,4) + 4\, y(t,4) = 78.83185307 + \frac{t}{-4}$$

$$4 \cdot \frac{\partial y}{\partial x}(t,8) + 0 \cdot y(t,8) = 50.26548246 + \frac{t^2}{-3} - 0.8 \cdot t$$

$$y(0,x) = 4 + 8 \cdot \sin\left(\frac{\pi}{2} x\right)$$

$$\frac{\partial y}{\partial t} = \frac{y_i^{k+1} - y_i^k}{0.025}, \quad \frac{\partial y}{\partial x} = \frac{y_{i+1}^{k+1} - y_{i-1}^{k+1}}{2 \cdot 0.8}$$

$$\frac{\partial^2 y}{\partial x^2} = \frac{y_{i+1}^{k+1} - 2y_i^{k+1} + y_{i-1}^{k+1}}{0.64}$$

$$y_0^1 = \frac{0.8 \cdot 78.83185307 - 5 y_1^1}{0.8 \cdot 4 - 5} \Rightarrow -1.8 \cdot y_0^1 + 5 y_1^1 = 63.060482456$$

$$y_5^1 = \frac{0.8 \cdot 50.2452741267 + 4 \cdot y_4^1}{0.8 \cdot 0 + 4} \Rightarrow -4 \cdot y_4^1 + y_5^1 = 40.1962193013$$

$$\frac{y_i^1 - y_i^0}{0.025} = 2.4 \cdot \frac{y_{i-1}^1 - 2y_i^1 + y_{i+1}^1}{0.64} + 0 + 6.3 \cdot y_i^1 + \frac{x_i + 3}{0.025 + 5}, \quad i = 1 \ldots 4$$

$$\Updownarrow$$

$$0.09375 \cdot y_{i-1}^1 - 1.03 \cdot y_i^1 + 0.09375 \cdot y_{i+1}^1 = \left\{ \begin{array}{c|l} i & \\ \hline 1 & -11.6472581005 \\ \hline 2 & -8.7450680880 \\ \hline 3 & 0.6554158492 \\ \hline 4 & 40.1962193013 \end{array} \right.$$

$$
\begin{array}{cccccc|c}
-1.8 & 5 & 0 & 0 & 0 & 0 & 63.060482456 \\
0.09375 & -1.03 & 0.09375 & 0 & 0 & 0 & -11.6472581005 \\
0 & 0.09375 & -1.03 & 0.09375 & 0 & 0 & -8.745068089 \\
0 & 0 & 0.09375 & -1.03 & 0.09375 & 0 & 0.6555158092 \\
0 & 0 & 0 & 0.09375 & -1.03 & 0.09375 & 3.557705864 \\
0 & 0 & 0 & 0 & -4 & 4 & 40.1962193013
\end{array}
$$

$$P_1 = \frac{-5}{-1.8} \simeq 2,(7) = \frac{-C_1}{-b_1}$$

$$P_i = \frac{-C_{i-1}}{(b_{i-1} + a_i \cdot P_{i-1})} \quad i = 2 \dots 6$$

$$P_2 = \frac{1.03}{(0.09375 + 0.9375 \cdot 2.(7))} =$$

$$= 0.1218191662$$

$$P_3 = 0.0920399$$

$$P_4 = 0.0917883674$$

$$P_5 = 0.0917862479$$

$$P_6 = 0$$

$$Q_1 = \frac{63.060482456}{-1.8} = -35.03369$$

$$= \frac{d_1}{b_1}$$

$$Q_i = \frac{d_{i-1} - a_{i-1} \cdot Q_{i-1}}{b_{i-1} + a_{i-1}} \quad i = 2 \dots 6$$

$$Q_2 = \frac{-11.6472581005 - 0.09375 \cdot (-35.033\dots}{-1.03 + 0.09375 \cdot 2.(7)} =$$

$$= 10.866734$$

$$Q_3 = 9.5857269092$$

$$Q_4 = 0.2380584409$$

$$Q_5 = -3.4613332107$$

$$Q_6 = 7.2534924726$$

$$Y'_6 = Q_6 = 7.2534924726$$

$$Y_i = Q_i + P_i \cdot Y_{i+1}, \quad i = 5 \dots 1$$

$$Y'_5 = -3.4613332107 + 0.0917862479 \cdot 7.2534924726 =$$
$$= -2.7955623528$$

$$Y'_4 = -0.0185416639$$

$$Y'_3 = 9.5840203355$$

$$Y'_2 = 12.0342521439$$

$$Y'_1 = -1.6051231871$$

остальные
шаги
аналогично.

Листинг программы:

```
#include <iostream>
#include <vector>
#include <iomanip>
#include <cmath>
#include <functional>
#include <string>
#include <exception>

using namespace std;
using f_RxRtoR = function<double(double, double)>; // сокращение названия типа по логике f : R -> R
using f_RtoR = function<double(double)>; // сокращение названия типа по логике f : R -> R

const int width = 14;
const string line(width, '-');

struct equasion{ // dy/dt = a1*d^2y/dx^2 + a2*dy/dx + a3*y + f(t,x)
  double a1, a2, a3;
  f_RxRtoR f;
};

struct edge_cond{ // fi1*dy/dx + fi2*y = f1(t)
  double fi1, fi2;
  f_RtoR f;
};

struct  region{
  double l, r, h;
};

//просто метод прогонки
vector<double> progonka(const vector<vector<double>>& mtrix){
  vector<double> P(mtrix.size()+1, 0.0),
             Q(mtrix.size()+1, 0.0),
             ans(mtrix.size(), 0.0);
  cout<<"Progonka:"<<endl;
  for(int i=1; i<P.size(); ++i){
       //Pi = -ci/(bi + ai * P(i-1))
       P[i]=-mtrix[i-1][2]/(mtrix[i-1][1] + mtrix[i-1][0]*P[i-1]);
       //cout<<"P["<<i+1<<"] = "<<(P[i])<<endl;
  }
  //cout<<endl;

  for(int i=1; i<Q.size(); ++i){
       //Qi = (di - ai*Q(i-1)) / (bi + ai * P(i-1))
       Q[i] = (mtrix[i-1][3] - mtrix[i-1][0]*Q[i-1])/(mtrix[i-1][1] +
mtrix[i-1][0]*P[i-1]);
       //cout<<"Q["<<i+1<<"] = "<<(Q[i])<<endl;
  }
  for(int i=0; i<Q.size(); ++i)
       cout<<"P["<<i+1<<"] =
"<<setprecision(width-4)<<fixed<<setw(width)<<(P[i])<<", Q["<<i+1<<"] =
"<<setprecision(width-4)<<fixed<<setw(width)<<(Q[i])<<endl;
  cout<<endl;
  //xn = qn
  ans[ans.size()-1] = Q[Q.size()-1];
```

```
  for(int i=ans.size()-1; i>0; --i)
       //xi = Qi + Pi * X(i+1)
       ans[i-1]=Q[i] + P[i]*ans[i];
  return ans;
}

void implict_grid_method(const equasion& eq, const edge_cond& cl, const edge_cond&
cr, const region& rx, const region& rt, f_RtoR cf, vector<vector<double>>& y){
  y.clear();
  y.push_back({});
  int nx = (rx.r - rx.l)/rx.h;
  int nt = (rt.r - rt.l)/rt.h;
  cout<<"x_i:\n";
  for(int j=0; j<=nx; j+=1)
       cout<<setprecision(width-4)<<fixed<<setw(width)<<rx.l+j*rx.h
<<((j!=nx)?",":"\n");
  cout<<endl;

  for(int k=0; k<=nt; k+=1){
       vector<vector<double>> matrix{};
       for(int i=0; i<=nx; i+=1){
       if(k==0){
       y[k].push_back(cf(rx.l+i*rx.h));
       continue;
       }
       if(i==0)
       matrix.push_back({
       0,
       (rx.h*cl.fi2 - cl.fi1),
       cl.fi1,
       cl.f(rt.l+rt.h*k)*rx.h});
       else if(i==nx)
       matrix.push_back({
       -cr.fi1,
       (rx.h*cr.fi2 + cr.fi1),
       0,
       cr.f(rt.l+rt.h*k)*rx.h});
       else
       matrix.push_back({
       (2.0*eq.a1 - rx.h*eq.a2)*rt.h/(2*rx.h*rx.h),
       (eq.a3*rt.h - 2.0*eq.a1*rt.h/(rx.h*rx.h) - 1.0),
       (2.0*eq.a1 + rx.h*eq.a2)*rt.h/(2*rx.h*rx.h),
       -y[k-1][i] - rt.h*eq.f(rt.l+rt.h*(k), rx.l+rx.h*i)});
       }
       cout<<"Step "<<k<<endl;

       if(k!=0){
       cout<<"Matrix:"<<endl;
       cout<<line<<'+'<<line<<'+'<<line<<'+'<<line<<'+'<<endl;

cout<<setw(width)<<'a'<<'|'<<setw(width)<<'b'<<'|'<<setw(width)<<'c'<<'|'<<setw(wid
th)<<'d'<<'|'<<endl;
       cout<<line<<'+'<<line<<'+'<<line<<'+'<<line<<'+'<<endl;
       for(auto l: matrix){
       for(auto d: l)
       cout<<setprecision(width-4)<<fixed<<setw(width)<<d<<'|';
       cout<<endl;
       }
       cout<<line<<'+'<<line<<'+'<<line<<'+'<<line<<'+'<<endl;
```

```
        cout<<endl;
        y.push_back(progonka(matrix));
        }
        cout<<"t_"<<k<<" = "<<rt.l+k*rt.h<<endl;
        cout<<"y_"<<k<<':'<<endl;
        for(int j=0; j<=nx; j+=1)
        cout<<setprecision(width-4)<<fixed<<setw(width)<<y.back()[j]
<<((j!=nx)?",":"\n");
        cout<<endl;
  }
}


int main(){
  //начальные условия
  equasion eq{2.4, 0, 6.3, [](double t, double x){return (x+3)/(t+5);}};
  region x{4, 8, 0.8},
      t{0, 0.1, 0.025};
  edge_cond cl{5, 4, [](double t){return 78.83185307 + t/(-4.0);}},
            cr{4, 0, [](double t){return 50.26548246 + t*t/(-3.0) - 0.8*t;}};
  f_RtoR cf = [](double x){return 4.0 + 8.0 * sin( M_PI/2 * x );};

  //условия примера из методички, решением которого является y = exp(-t)*sin(x)
  // equasion eq{1, 0, 0, [](double t, double x){return 0;}};
  // region x{0, M_PI, M_PI/10.0}, t{0, 2, 0.1};
  // edge_cond cl{0, 1, [](double t){return 0;}},
  //           cr{0, 1, [](double t){return 0;}};
  // f_RtoR cf = [](double x){return sin( x );};

  vector<vector<double>> y;

  implict_grid_method(eq, cl, cr, x,t, cf, y);
  cout<<"Table of results"<<endl;
  for(auto l: y){
      for(auto d: l)
        cout<<d<<' ';
      cout<<endl;
  }


  cout<<"Res for T_end = "<<t.r<<" :"<<endl;
  for(int i=0; i<=int((x.r-x.l)/x.h); i+=1)

cout<<"("<<setprecision(width-4)<<fixed<<setw(width)<<x.l+i*x.h<<","<<setprecision(
width-4)<<fixed<<setw(width)<<y.back()[i] <<((i!=int((x.r-x.l)/x.h))?"), ":")\n");

  return 0;
}
```

Демонстрация работы:

```
schizophrenia@home:~/labs/3kurs/Chislaki/kp$ g++
parabl_dif_eq_implict_first_order.cpp && ./a.out
x_i:
  4.0000000000,  4.8000000000,  5.6000000000,  6.4000000000,  7.2000000000,
8.0000000000

Step 0
t_0 = 0.0000000000
y_0:
```

```
  4.0000000000, 11.6084521304,  8.7022820183, -0.7022820183, -3.6084521304,
4.0000000000

Step 1
Matrix:
--------------+--------------+--------------+--------------+
             a|             b|             c|             d|
--------------+--------------+--------------+--------------+
  0.0000000000| -1.8000000000|  5.0000000000| 63.0604824560|
  0.0937500000| -1.0300000000|  0.0937500000|-11.6472581005|
  0.0937500000| -1.0300000000|  0.0937500000| -8.7450680880|
  0.0937500000| -1.0300000000|  0.0937500000|  0.6555158492|
  0.0937500000| -1.0300000000|  0.0937500000|  3.5577058617|
 -4.0000000000|  4.0000000000|  0.0000000000| 40.1962193013|
--------------+--------------+--------------+--------------+

Progonka:
P[1] =   0.0000000000, Q[1] =   0.0000000000
P[2] =   2.7777777778, Q[2] = -35.0336013644
P[3] =   0.1218191662, Q[3] =  10.8667347776
P[4] =   0.0920399481, Q[4] =   9.5857269092
P[5] =   0.0917883674, Q[5] =   0.2380584409
P[6] =   0.0917862479, Q[6] =  -3.4613332107
P[7] =  -0.0000000000, Q[7] =   7.2534924726

t_1 = 0.0250000000
y_1:
 -1.6051231871, 12.0342521439,  9.5840203355, -0.0185416634, -2.7955623528,
7.2534924726

Step 2
Matrix:
--------------+--------------+--------------+--------------+
             a|             b|             c|             d|
--------------+--------------+--------------+--------------+
  0.0000000000| -1.8000000000|  5.0000000000| 63.0554824560|
  0.0937500000| -1.0300000000|  0.0937500000|-12.0728660052|
  0.0937500000| -1.0300000000|  0.0937500000| -9.6265945929|
  0.0937500000| -1.0300000000|  0.0937500000| -0.0279929901|
  0.0937500000| -1.0300000000|  0.0937500000|  2.7450673033|
 -4.0000000000|  4.0000000000|  0.0000000000| 40.1797193013|
--------------+--------------+--------------+--------------+

Progonka:
P[1] =   0.0000000000, Q[1] =   0.0000000000
P[2] =   2.7777777778, Q[2] = -35.0308235867
P[3] =   0.1218191662, Q[3] =  11.4201099651
P[4] =   0.0920399481, Q[4] =  10.5021065061
P[5] =   0.0917883674, Q[5] =   0.9913784730
P[6] =   0.0917862479, Q[6] =  -2.5965723206
P[7] =  -0.0000000000, Q[7] =   8.2011062783

t_2 = 0.0500000000
y_2:
  0.2710814445, 12.7086858112, 10.5777759456,  0.8221369198, -1.8438235470,
8.2011062783

Step 3
Matrix:
```

```
--------------+--------------+--------------+--------------+
             a|             b|             c|             d|
--------------+--------------+--------------+--------------+
  0.0000000000| -1.8000000000|  5.0000000000| 63.0504824560|
  0.0937500000| -1.0300000000|  0.0937500000|-12.7471094565|
  0.0937500000| -1.0300000000|  0.0937500000|-10.6201404776|
  0.0937500000| -1.0300000000|  0.0937500000| -0.8684423385|
  0.0937500000| -1.0300000000|  0.0937500000|  1.7935772416|
 -4.0000000000|  4.0000000000|  0.0000000000| 40.1628859680|
--------------+--------------+--------------+--------------+

Progonka:
P[1] =   0.0000000000, Q[1] =   0.0000000000
P[2] =   2.7777777778, Q[2] = -35.0280458089
P[3] =   0.1218191662, Q[3] =  12.2965632857
P[4] =   0.0920399481, Q[4] =  11.5581982819
P[5] =   0.0917883674, Q[5] =   1.9111791312
P[6] =   0.0917862479, Q[6] =  -1.5805877746
P[7] =  -0.0000000000, Q[7] =   9.3151350081

t_3 = 0.0750000000
y_3:
  3.0976630908, 13.7252552039, 11.7279732127,  1.8445787324, -0.7255864839,
9.3151350081

Step 4
Matrix:
--------------+--------------+--------------+--------------+
             a|             b|             c|             d|
--------------+--------------+--------------+--------------+
  0.0000000000| -1.8000000000|  5.0000000000| 63.0454824560|
  0.0937500000| -1.0300000000|  0.0937500000|-13.7634904980|
  0.0937500000| -1.0300000000|  0.0937500000|-11.7701300755|
  0.0937500000| -1.0300000000|  0.0937500000| -1.8906571638|
  0.0937500000| -1.0300000000|  0.0937500000|  0.6755864839|
 -4.0000000000|  4.0000000000|  0.0000000000| 40.1457193013|
--------------+--------------+--------------+--------------+

Progonka:
P[1] =   0.0000000000, Q[1] =   0.0000000000
P[2] =   2.7777777778, Q[2] = -35.0252680311
P[3] =   0.1218191662, Q[3] =  13.6175917099
P[4] =   0.0920399481, Q[4] =  12.8087988205
P[5] =   0.0917883674, Q[5] =   3.0267956323
P[6] =   0.0917862479, Q[6] =  -0.3836169695
P[7] =  -0.0000000000, Q[7] =  10.6283491448

t_4 = 0.1000000000
y_4:
  7.2316630954, 15.2124952055, 13.0923855841,  3.0811269403,  0.5919193195,
10.6283491448

Table of results
4.0000000000 11.6084521304 8.7022820183 -0.7022820183 -3.6084521304 4.0000000000
-1.6051231871 12.0342521439 9.5840203355 -0.0185416634 -2.7955623528 7.2534924726
0.2710814445 12.7086858112 10.5777759456 0.8221369198 -1.8438235470 8.2011062783
3.0976630908 13.7252552039 11.7279732127 1.8445787324 -0.7255864839 9.3151350081
7.2316630954 15.2124952055 13.0923855841 3.0811269403 0.5919193195 10.6283491448
Res for T_end = 0.1000000000 :
```

(4.0000000000, 7.2316630954), (4.8000000000, 15.2124952055), (5.6000000000, 13.0923855841), (6.4000000000, 3.0811269403), (7.2000000000, 0.5919193195), (8.0000000000, 10.6283491448)

График решения: