БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Лабораторная работа №3

Решение второй краевой задачи для уравнения теплопроводности

**Преподаватель:** Полевиков Виктор Кузьмич

**Студент:** Иваненко Даниил

3 курс 9 группа

**2021 г.**

**Исходные данные:**

**Постановка задачи:**

1. Построить Явную и Неявную разностную схему.
2. Исследовать порядок аппроксимации. Для внутренних узлов
3. Увеличить порядок аппроксимации на границах по x.
4. Решение изобразить на графике при t=1.

**Краткая теория**

Для задачи в качестве краевой схемы возьмём

=

Из них можно выразить

Найдем порядок аппроксимации во внутренних узлах:

Начальное условие . Граничное условие на правом конце . Граничное условие на левом конце

Повысим порядок аппроксимации граничного условия на левом конце, не расширяя шаблона:

Погрешность аппроксимации .

**Явная схема** с порядком аппроксимации :

Для нахождения проверим условия на коэффициенты во внутренних узлах из формулировки принципа максимума:

Условия выполняются:

Проверим условия на границе:

Условия для коэффицентов выполняются:

Из полученного можно заключить, что данная разностная схема будет устойчивой при , согласно принципу максимума.

**Чисто неявная схема** с порядком аппроксимации :

На каждом слое имеем СЛАУ с трёхдиагональной матрицей:

Условия строгого диагонального преобладания выполнены. Система решается методом прогонки, который для нее абсолютно устойчив. Чтобы решить разностную задачу полностью, необходимо применить прогонку раз, т.е. решить с ее помощью СЛАУ относительно

**Листинг программы**

#define \_USE\_MATH\_DEFINES#include <iostream>#include <cmath>using namespace std;void Metoda(double\*\* matrix, double\* x, double\* f, int dlina){double delitel = 1;for (int prohodov = 0;prohodov < dlina;prohodov++){for (int i = prohodov;i < dlina - 1;i++){if (matrix[i + 1][prohodov] != 0)delitel = matrix[i + 1][prohodov] / matrix[prohodov][prohodov];elsebreak;for (int j = prohodov;j < dlina;j++)matrix[i + 1][j] -= delitel \* matrix[prohodov][j];f[i + 1] -= delitel \* f[prohodov];}}for (int t = dlina - 1;t >= 0;t--) {x[t] = f[t] / matrix[t][t];for (int i = 0;i < dlina;i++)f[i] -= matrix[i][t] \* x[t];}}double Function\_of\_find\_norm(double\* f) {double sum = 0;for (int i = 0;i < 6;i++)sum += pow(f[i], 2);return pow(sum, 0.5);}void Function\_of\_find\_b(double\*x,double\*f,int i\_1,double\* b) {for (int i = 0;i < i\_1;i++)b[i] = 0;for (int i = 0;i < i\_1;i++)for (int j = 0;j < 6;j++)b[i] += f[j] \* pow(x[j], i);}void Function\_of\_find\_G(double\*\* G, double\* x, int i\_0) {for (int i = 0;i < i\_0;i++)for (int k = 0;k < i\_0;k++)G[i][k] = 0;for (int i = 0;i < i\_0;i++)for (int k = 0;k < i\_0;k++)for (int j = 0;j < 6;j++)G[i][k] += pow(x[j], i) \* pow(x[j], k);}void printmatrix2(double\*\* matrix, int dlina){for (int i = 0;i < dlina;i++){for (int j = 0;j < dlina;j++)printf(" %0.5f", matrix[i][j]);cout << "\n";}}double Function\_of\_find\_delta(double\* x,double\*f,double\*a,int i\_0) {double sum = 0;double lsum = 0;for (int i = 0;i < 6;i++) {or (int j = 0;j < i\_0;j++)lsum += a[j] \* pow(x[i], j);lsum -= f[i];lsum = pow(lsum, 2);sum += lsum;lsum = 0;}return pow(sum,0.5);}double Function\_of\_find\_delta\_d(double\* f, double delta) {return delta / Function\_of\_find\_norm(f);}int main() {double delta ;double delta\_D ;double\* x = new double[6];double\* f = new double[6];or (int i = 0;i < 6;i++) {x[i] = i \* 0.2;f[i] = pow(sin(M\_PI\*x[i])/(1+x[i]),2);//f[i] = pow(abs(cos(M\_PI \* x[i])) / (1 + x[i]), 0.5);}cout << "X:\n";for(int i=0;i<6;i++)printf(" %0.2f", x[i]);cout << "\nF:\n";for (int i = 0;i < 6;i++)printf(" %0.2f", f[i]);for (int i = 1;i <= 3;i++) {cout << "\n\t\t\t\tN: " << i<<"\n";delta = 0;delta\_D = 0;double\* a = new double[i];double\* b = new double[i];double\*\* G = new double\*[i];for (int k = 0;k<i;k++)G[k] = new double[i];Function\_of\_find\_b(x, f, i, b);cout << "\nb:\n";for (int k = 0;k < i;k++)printf(" %0.2f", b[k]);Function\_of\_find\_G(G, x, i);cout << "\nG:\n";printmatrix2(G, i);MetodGaussa(G, a, b, i);cout << "\na:\n";for (int k = 0;k < i;k++)printf(" %0.2f", a[k]);delta=Function\_of\_find\_delta(x, f, a, i);delta\_D = Function\_of\_find\_delta\_d(f, delta);cout << "\nDelta:" << delta << " Delta\_D:" << delta\_D << "\n";}return 0;} double\* x, double\* f, int dlina){double delitel = 1;for (int prohodov = 0;prohodov <MCHA\_NE\_NUJON\_ohodov;i < dlina - 1;i++){if (matrix[i + 1][prohodov] != 0)delitel = matrix[i + 1][prohodov] / matrix[prohodov][prohodov];elsebreak;for (int j = prohodov;j < dlina;j++)matrix[i + 1][j] -= delitel \* matrix[prohodov][j];f[i + 1] -= delitel \* f[prohodov];}}for (int t = dlina - 1;t >= 0;t--) {x[t] = f[t] / matrix[t][t];for (int i = 0;i < dlina;i++)f[i] -= matrix[i][t] \* x[t];}}double Function\_of\_find\_norm(double\* f) {double sum = 0;for (int i = 0;i < 6;i++)sum += pow(f[i], 2);return pow(sum, 0.5);}void Function\_of\_find\_b(double\*x,double\*f,int i\_1,double\* b) {for (int i = 0;i < i\_1;i++)b[i] = 0;for (int i = 0;i < i\_1;i++)for (int j = 0;j < 6;j++)b[i] += f[j] \* pow(x[j], i);}void Function\_of\_find\_G(double\*\* G, double\* x, int i\_0) {for (int i = 0;i < i\_0;i++)for (int k = 0;k < i\_0;k++)G[i][k] = 0;for (int i = 0;i < i\_0;i++)for (int k = 0;k < i\_0;k++)for (int j = 0;j < 6;j++)G[i][k] += pow(x[j], i) \* pow(x[j], k);}void printmatrix2(double\*\* matrix, int dlina){for (int i = 0;i < dlina;i++){for (int j = 0;j < dlina;j++)printf(" %0.5f", matrix[i][j]);cout << "\n";}}double Function\_of\_find\_delta(double\* x,double\*f,double\*a,int i\_0) {double sum = 0;double lsum = 0;for (int i = 0;i < 6;i++) {or (int j = 0;j < i\_0;j++)lsum += a[j] \* pow(x[i], j);lsum -= f[i];lsum = pow(lsum, 2);sum += lsum;lsum = 0;}return pow(sum,0.5);}double Function\_of\_find\_delta\_d(double\* f, double delta) {return delta / Function\_of\_find\_norm(f);}int main() {double delta ;double delta\_D ;double\* x = new double[6];double\* f = new double[6];or (int i = 0;i < 6;i++) {x[i] = i \* 0.2;f[i] = pow(sin(M\_PI\*x[i])/(1+x[i]),2);//f[i] = pow(abs(cos(M\_PI \* x[i])) / (1 + x[i]), 0.5);}cout << "X:\n";for(int i=0;i<6;i++)printf(" %0.2f", x[i]);cout << "\nF:\n";for (int i = 0;i < 6;i++)printf(" %0.2f", f[i]);for (int i = 1;i <= 3;i++) {cout << "\n\t\t\t\tN: " << i<<"\n";delta = 0;delta\_D = 0;double\* a = new double[i];double\* b = new double[i];double\*\* G = new double\*[i];for (int k = 0;k<i;k++)G[k] = new double[i];Function\_of\_find\_b(x, f, i, b);cout << "\nb:\ double\* x, double\* f, int dlina){double delitel = 1;for (int prohodov = 0;prohodov < dlina;prohodov++){for (int i = prohodov;i < dlina - 1;i++){if (matrix[i + 1][prohodov] != 0)delitel = matrix[i + 1][prohodov] / matrix[prohodov][prohodov];elsebreak;for (int j = prohodov;j < dlina;j++)matrix[i + 1][j] -= delitel \* matrix[prohodov][j];f[i + 1] -= delitel \* f[prohodov];}}for (int t = dlina - 1;t >= 0;t--) {x[t] = f[t] / matrix[t][t];for (int i = 0;i < dlina;i++)f[i] -= matrix[i][t] \* x[t];}}double Function\_of\_find\_norm(double\* f) {double sum = 0;for (int i = 0;i < 6;i++)sum += pow(f[i], 2);return pow(sum, 0.5);}void Function\_of\_find\_b(double\*x,double\*f,int i\_1,double\* b) {for (int i = 0;i < i\_1;i++)b[i] = 0;for (int i = 0;i < i\_1;i++)for (int j = 0;j < 6;j++)b[i] += f[j] \* pow(x[j], i);}void Function\_of\_find\_G(double\*\* G, double\* x, int i\_0) {for (int i = 0;i < i\_0;i++)for (int k = 0;k < i\_0;k++)G[i][k] = 0;for (int i = 0;i < i\_0;i++)for (int k = 0;k < i\_0;k++)for (int j = 0;j < 6;j++)G[i][k] += pow(x[j], i) \* pow(x[j], k);}void printmatrix2(double\*\* matrix, int dlina){for (int i = 0;i < dlina;i++){for (int j = 0;j < dlina;j++)printf(" %0.5f", matrix[i][j]);cout << "\n";}}double Function\_of\_find\_delta(double\* x,double\*f,double\*a,int i\_0) {double sum = 0;double lsum = 0;for (int i = 0;i < 6;i++) {or (int j = 0;j < i\_0;j++)lsum += a[j] \* pow(x[i], j);lsum -= f[i];lsum = pow(lsum, 2);sum += lsum;lsum = 0;}return pow(sum,0.5);}double Function\_of\_find\_delta\_d(double\* f, double delta) {return delta / Function\_of\_find\_norm(f);}int main() {double delta ;double delta\_D ;double\* x = new double[6];double\* f = new double[6];or (int i = 0;i < 6;i++) {x[i] = i \* 0.2;f[i] = pow(sin(M\_PI\*x[i])/(1+x[i]),2);//f[i] = pow(abs(cos(M\_PI \* x[i])) / (1 + x[i]), 0.5);}cout << "X:\n";for(int i=0;i<6;i++)printf(" %0.2f", x[i]);cout << "\nF:\n";for (int i = 0;i < 6;i++)printf(" %0.2f", f[i]);for (int i = 1;i <= 3;i++) {cout << "\n\t\t\t\tN: " << i<<"\n";delta = 0;delta\_D = 0;double\* a = new double[i];double\* b = new double[i];double\*\* G = new double\*[i];for (int k = 0;k<i;k++)G[k] = new double[i];Function\_of\_find\_b(x, f, i, b);cout << "\nb:\

**Результаты**

Первые и последние 5 значений в узлах точного и приближённого решения:

|  |  |  |  |
| --- | --- | --- | --- |
| x | Точное | ЯРС | НРС |
| 0 | 0,909297452 | 0,909296537 | 0,909301956 |
| 0,1 | 1,818686356 | 1,818687454 | 1,818696382 |
| 0,2 | 2,728347523 | 2,728351455 | 2,728349374 |
| 0,3 | 3,638463344 | 3,638467482 | 3,638465353 |
| 0,4 | 4,549217528 | 4,549220997 | 4,549220194 |
| 0,96 | 125,092786786 | 125,092780384 | 125,092787896 |
| 0,97 | 127,422667251 | 127,422661743 | 127,422662657 |
| 0,98 | 129,798426248 | 129,798425188 | 129,79843895 |
| 0,99 | 132,221451743 | 132,221450478 | 132,221451701 |
| 1 | 134,693178415 | 134,693178415 | 134,693178415 |