Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Методы численного анализа»

ОТЧЕТ

к лабораторной работе №12
на тему:

«**РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ МЕТОДОМ РАЗНОСТНЫХ АППРОКСИМАЦИЙ.**»
БГУИР 1-40 04 01

Выполнил студент группы 253505
Сенько Никита Святославович

Проверил доцент кафедры информатики
АНИСИМОВ Владимир Яковлевич

Минск 2024

СОДЕРЖАНИЕ

Цель:

1. Изучить метод разностных аппроксимаций, составить алгоритм метода и программу их реализации
2. Составить алгоритм решения краевых задач методами, применимыми для организации вычислений на ПЭВМ.

Задание

Задача 1. Составить разностную схему и получить численное решение краевой задачи с точностью до 10^{-3}:

$$ay'' + (1 + bx^2)y = -1, \qquad -1 \le x \le 1,$$

Исходные данные:

$$a = \sin(k), b = \cos(k),$$

где k – номер варианта.

Граничные условия выбрать однородными:

$$y(-1) = 0,$$
$$y(1) = 0.$$

Задача 2. Найти приближенное решение краевой задачи методом конечных разностей:

$$\begin{cases} u'' + p(x)u' + q(x)u = f(x), \quad x \in (a,b), \\ u(a) = UA, \quad u(b) = UB \end{cases}$$

с заданной точностью ε и построить его график. Исходные данные указаны в

Задача 3. Методом конечных разностей найти приближенное решение указанной в индивидуальном варианте краевой задачи (см. табл. 2.4) с точностью ε (табл. 2.2) и построить его график. Решение системы разностных уравнений найти, используя метод прогонки.

Задача 4. Методом конечных разностей найти приближенное решение краевой задачи с тремя верными значащими цифрами. Решение системы разностных уравнений найти, используя метод прогонки. Исходные данные указаны в табл. 2.3.

$$\begin{cases} -(k(x)u')' + q(x)u = f(x), \quad x \in (a,b), \\ -k(a)u'(a) + 0.5u(a) = 0, \\ \ k(b)u'(b) + 0.5u(b) = 0. \end{cases}$$

Программная реализация

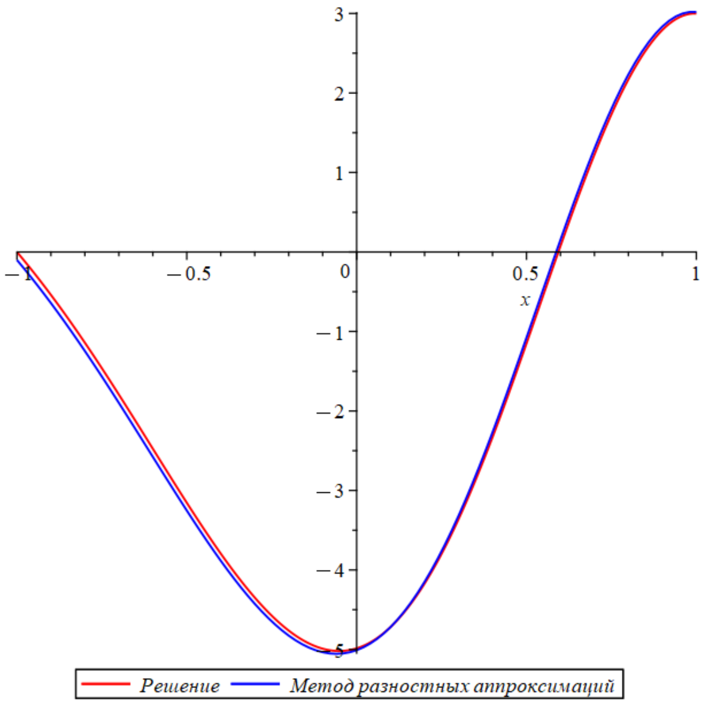
Метод разностных аппроксимаций y''+p(x)y'+q(x)y=f(x)

$difference\_method := \mathbf{proc}(p, q, f, h, l, r, L, R)$
local $a, b, c, d, i, n, x\_arr, xk$;
$a := Array([\,])$;
$b := Array([\,])$;
$c := Array([\,])$;
$d := Array([\,])$;
$x\_arr := Array([\,])$;
$n := \text{floor}\left(\frac{r-l}{h}\right)$;
$xk := l$;
for i **from** 1 **to** $n+1$ **do**
$ArrayTools\text{:-}Append\left(a, 1 - \frac{p(xk)}{2} \cdot h\right)$;
$ArrayTools\text{:-}Append\left(b, -2 + q(xk) \cdot h^2\right)$;
$ArrayTools\text{:-}Append\left(c, 1 + \frac{p(xk)}{2} \cdot h\right)$;
$ArrayTools\text{:-}Append\left(d, f(xk) \cdot h^2\right)$;
$ArrayTools\text{:-}Append(x\_arr, xk)$;
$xk := xk + h$;
end do;
return $[x\_arr, running(a, b, c, d, L, R)]$
end proc:

алгоритм Томаса

$running := \mathbf{proc}(a, b, c, f, L, R)$
local $i, n, x\_arr, x, alpha\_arr, beta\_arr$;
$n := numelems(a)$;
$alpha\_arr := Array\left(\left[-\frac{c[1]}{b[1]}\right]\right)$;
$beta\_arr := Array\left(\left[\frac{f[1]}{b[1]}\right]\right)$;
$f[1] := f[1] - a[1] \cdot L$;
$f[n] := f[n] - c[n] \cdot R$;
$a[1] := 0$;
$c[n] := 0$;
for i **from** 2 **to** n **do**
$ArrayTools\text{:-}Append\left(alpha\_arr, -\frac{c[i-1]}{a[i-1] \cdot alpha\_arr[i-1] + b[i-1]}\right)$;
$ArrayTools\text{:-}Append\left(beta\_arr, \frac{f[i-1] - a[i-1] \cdot beta\_arr[i-1]}{a[i-1] \cdot alpha\_arr[i-1] + b[i-1]}\right)$;
end do;
$x\_arr := Array([\,])$;
for i **from** 1 **to** n **do**
$ArrayTools\text{:-}Append(x\_arr, cat(x, i))$;
end do;
$x\_arr[n] := \frac{-a[n] \cdot beta\_arr[n] + f[n]}{a[n] \cdot alpha\_arr[n] + b[n]}$;
for i **from** $n-1$ **to** 1 **by** -1 **do**
$x\_arr[i] := alpha\_arr[i+1] \cdot x\_arr[i+1] + beta\_arr[i+1]$;
end do;
return x_arr
end proc:

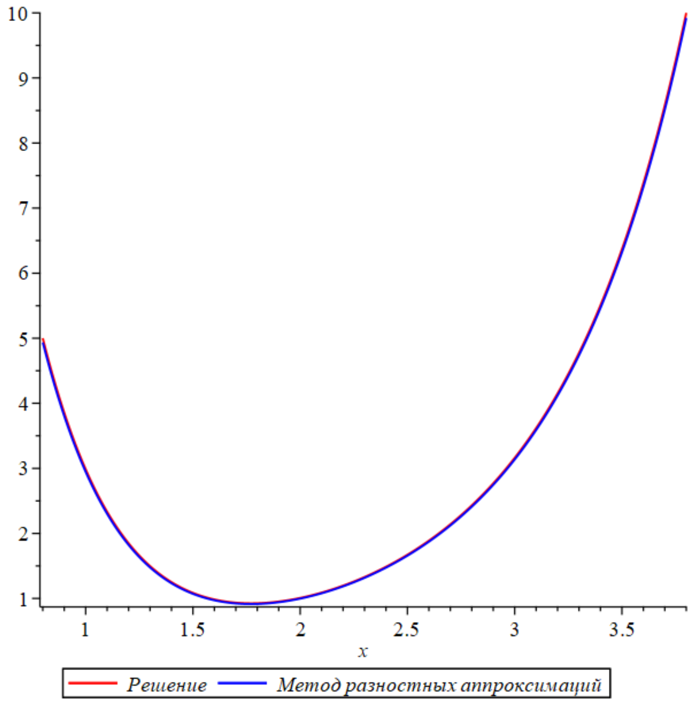
```
chebyshev_norm := proc(x_arr, y_arr, func)
 local res, i, temp;
 res := 0;
 for i from 1 to numelems(x_arr) do
  temp := abs(y_arr[i] - rhs(evalf(subs(x=x_arr[i], sol))));
  if temp > res then
  res := temp;
  end if
 end do;
 return res;
 end proc:
```




```
> chebyshev_norm(res[1], evalf(res[2]), sol)
```

0.104694423

```
> plot1 := plot(rhs(sol), x = l..r) :
> res := difference_method(p, q, f, h, l, r, L, R) :
> plot2 := plots:-pointplot([seq([convert(res[1][i], float), convert(res[2][i], float)], i = 1..numelems(res[1]))], connect = true) :
> plots[display](plot1, plot2, color = [red, blue], legend = [Решение, Метод разностных аппроксимаций])
```



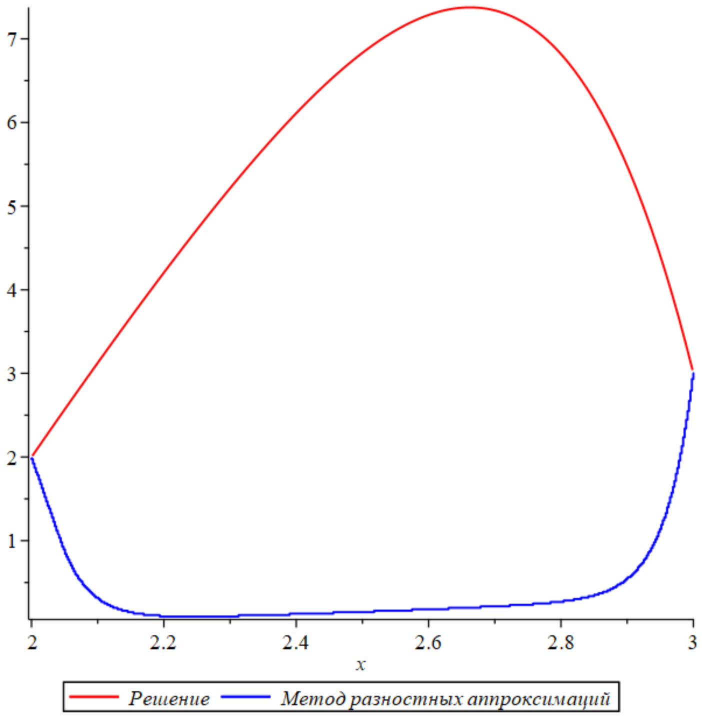

```
> chebyshev_norm(res[1], res[2], sol)
```

0.077020123

#4

```
>
```

res := *difference_method*(*p*, *q*, *f*, *h*, *l*, *r*, *L*, *R*) :
plot2 := *plots*:-*pointplot*([*seq*([*convert*(*res*[1][*i*], *float*), *convert*(*res*[2][*i*], *float*)], *i* = 1 ..*numelems*(*res*[1]))], *connect* = *true*) :
plots[*display*](*plot1*, *plot2*, *color* = [*red*, *blue*], *legend* = [*Решение*, *Метод разностных аппроксимаций*])

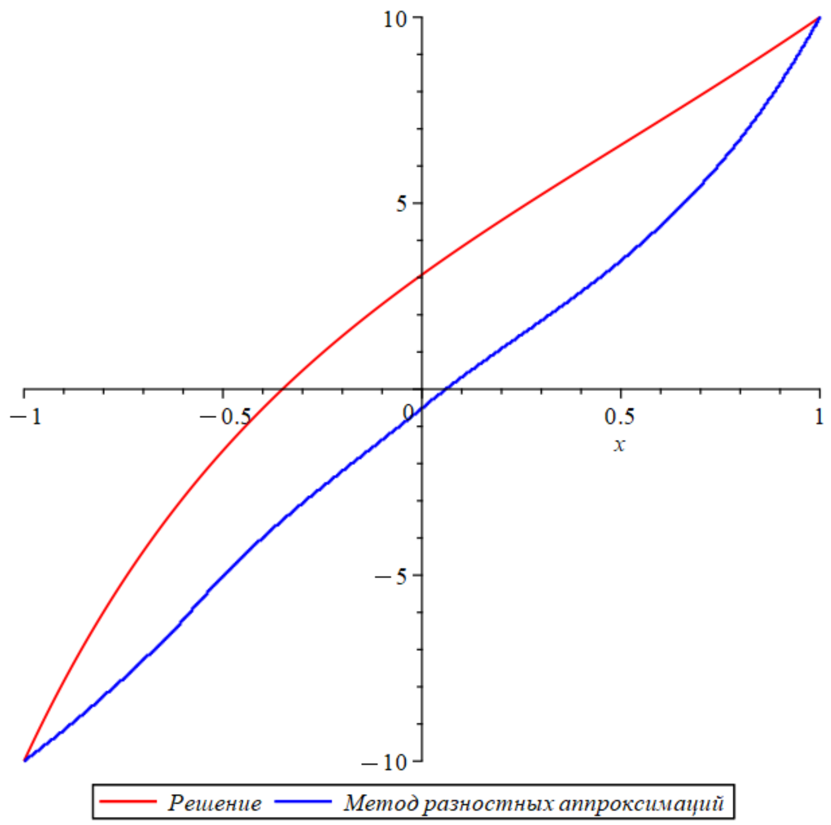


chebyshev_norm(*res*[1], *res*[2], *sol*)

7.178039157

#5 (1)

ay](*plot1*, *plot2*, *color* = [*red*, *blue*], *legend* = [*Решение*, *Метод разностных аппроксимаций*])



norm(*res*[1], *res*[2], *sol*)

3.648654868

(*y*(*x*), *x*$2) + 123456 · *x* · *y*(*x*) = 999999

Вывод

1. **Чувствительность метода к параметрам**:
   - Разностные схемы могут быть чувствительны к значениям коэффициентов p и q, а также к характеру функции f(x). Разные значения этих параметров могут существенно влиять на устойчивость и точность численного решения.
2. **Особенности данных**:
   - Графики, которые аппроксимировались нормально, вероятно, соответствуют случаям, где разностная схема эффективно справляется с решением. Это может происходить при определенных сочетаниях коэффициентов p и q или при определенном виде функции f(x).
   - Графики, которые не аппроксимировались нормально, могут быть связаны с более сложными или менее подходящими для данной разностной схемы случаями,