Отчет по заданию "Применение метода разностной прогонки к решению линейного дифференциального уравнения 2-го порядка" студентки 491 группы Апетян Арины

1 Постановка задачи

Дано линейное неоднородное дифференциальное уравнение 2-го порядка:

$$y'' + p(x)y' + q(x)y = f(x) \tag{1}$$

с граничными условиями (типа III):

$$y'(a) = \alpha y(a) + A; y'(b) = \beta y(b) + B \tag{2}$$

в частности:

$$\begin{array}{ll} p(x) = 1 + x & y'(0) = 1 \\ q(x) = \frac{x}{1+sin\frac{\pi}{2}x} & y'(1) = -0.3y(1) + 2 \\ f(x) = x(1-x) & \end{array}$$

Необходимо численно решить уравнение (1) с граничными условиями (2), используя разностный метод.

2 Используемые формулы

- сетка: $h = \frac{b-a}{n}$; $x_k = a + kh$; $k = 0, ..., n$
- приближенные значения в точках сетки: $y_k \approx y(x_k)$
- формулы численного дифференцирования:

$$y'(x) = \frac{y(x+h)-y(x)}{h} + R;$$
$$y'(x) = \frac{y(x)-y(x-h)}{h} + R;$$
$$y'(x) = \frac{y(x+h)-y(x-h)}{2h} + R;$$
$$y''(a) = \frac{y(x+h)-2y(x)+y(x-h)}{h^2} + R$$

- эквивалентная система:

$$\begin{cases} -b_0y_0 + c_0y_1 & = & d_0, \\ a_ky_{k-1} - b_ky_k + c_ky_{k+1} & = & d_k, \quad k = 1, ..., n-1 \\ a_ny_{n-1} - b_ny_n & = & d_n. \end{cases}$$

здесь:

$$\begin{array}{lllllllll} a_k & = & \frac{1}{h^2} - \frac{p_k}{2h} & a_n & = & \frac{p_n}{2} - \frac{1}{h} & & & \\ b_k & = & \frac{2}{h^2} - q_k & b_0 & = & \frac{1}{h} + \frac{p_0}{2} - \frac{h}{2}q_0 + \alpha & b_n & = & \frac{p_n}{2} - \frac{1}{h} + \frac{h}{2}q_n + \beta \\ c_k & = & \frac{1}{h^2} + \frac{p_k}{2h} & c_0 & = & \frac{1}{h} + \frac{p_0}{2} & & & \\ d_k & = & f_k & d_0 & = & A + \frac{h}{2}f_0 & d_n & = & B - \frac{h}{2}f_n \end{array}$$

- прогоночные соотношения:

$$y_0 = \alpha_0y_1 + \beta_0; \ \alpha_0 = \frac{c_0}{b_0}; \ \beta_0 = -\frac{d_0}{b_0}$$

$$\begin{array}{lll} y_{k-1} & = & \alpha_{k-1}y_k + \beta_{k-1} \\ \alpha_k & = & \frac{c_k}{b_k - a_k\alpha_{k-1}} \\ \beta_k & = & \frac{a_k\beta_{k-1} - d_k}{b_k - a_k\alpha_{k-1}} \end{array}$$

3 Текст программы

```
program thomasalgorithm
implicit none
integer n,k
real(8) ax,bx,h,alfx,betx,aa,bb,pi
real(8), allocatable, dimension(:) :: x,y,a,b,c,d,f,p,q,alf,bet
open(unit=1, file='results.dat')
n=10
pi=4*atan(1.0)
ax=0.0; bx=1.0; h=(bx-ax)/n;
alfx=0; betx=-0.3; aa=1.0; bb=2.0    ! the boundary conditions data
allocate(x(0:n),y(0:n), a(0:n), b(0:n), c(0:n), &
d(0:n), alf(0:n),bet(0:n), f(0:n),q(0:n),p(0:n))
  do k=0,n
    x(k) = ax + k*h
    f(k) = x(k)*(1-x(k))              ! defining
    q(k) = x(k)/(1+sin(pi/2*x(k)))    ! the
    p(k) = 1+x(k)                     ! equation
      a(k) = 1/h**2-p(k)/(2*h)
      b(k) = 2/h**2-q(k)              ! solving
      c(k) = 1/h**2+p(k)/(2*h)        ! the
      d(k) = f(k)                     ! system
  enddo
      c(0) = 1/h+p(0)/2
      b(0) = 1/h+p(0)/2-h/2*q(0)+alfx
      d(0) = h/2*f(0) + aa
        a(n) = -1/h+p(n)/2
        b(n) = -1/h+p(n)/2+h/2*q(n)+betx
        d(n) = -h/2*f(n) + bb
          alf(0) = c(0)/b(0); bet(0) = - d(0)/b(0)
            do k=1,n
              alf(k) = c(k)/(b(k)-a(k)*alf(k-1))
              bet(k) = (a(k)*bet(k-1)-d(k))/(b(k)-a(k)*alf(k-1))
            enddo
              y(n) = bet (n)
                do k=n-1,0,-1
                  y(k) = alf(k)*y(k+1)+bet(k)
                enddo
                  write(1,*) "a, b: "
                    do k=0,n
                      write(1,*) a(k),b(k)
                    enddo
                  write(1,*) "c, d: "
                    do k=0,n
                      write(1,*) c(k),d(k)      ! output
                    enddo
                  write(1,*) "alf, bet: "
                    do k=0,n
                      write(1,*) alf(k), bet(k)
                    enddo
                  write(1,*) "x, y: "
                    do k=0,n
                      write(1,*) x(k),y(k)
                    enddo
deallocate(x,y,a,b,c,d,f,p,q,alf,bet)
close(1)
end
```

4 Результаты

Ниже представлены результаты работы программы для сетки, состоящей из $n = 10, 20, 40$ узлов. Проверка сходимости осуществляется по разности значений в соответствующих точках.

	n = 10	n = 20	n = 40
y_0	12.946469	12.893340	12.880149
y_1	13.041707	12.987060	12.973490
y_2	13.117178	13.061323	13.047453
y_3	13.166708	13.109942	13.095845
y_4	13.186656	13.129264	13.115012
y_5	13.175189	13.117445	13.103104
y_6	13.131752	13.073908	13.059542
y_7	13.056643	12.998933	12.984601
y_8	12.950692	12.893326	12.879079
y_9	12.814988	12.758151	12.744034
y_{10}	12.650662	12.594512	12.580565