Аппроксимация решения краевой задачи

1 Метод разностной прогонки. Отчет

1.1 Условие задачи

Условие задачи в общем виде:

$$\begin{cases} y''(x) + p(x)y'(x) + q(x)y(x) = f(x) \\ \alpha_1 y(a) + \alpha_2 y'(a) = A \\ \beta_1 y(b) + \beta_2 y'(b) = B \end{cases}$$

Данные для моей задачи:

$$p(x) = x + \frac{3}{2}, \quad q(x) = \frac{x+1}{x+3}, \quad f(x) = \frac{1}{\sqrt{x^2+1}}$$

$$[a, b] = [0, 1]$$

$$\alpha_1 = 1, \quad \alpha_2 = 0, \quad A = 1.3$$

$$\beta_1 = 0.76, \quad \beta_2 = 1, \quad B = 0.23$$

С такими данными условие выглядит следующим образом:

$$y''(x) + (x + \frac{3}{2})\, y'(x) + \frac{x+1}{x+3}\, y(x) = \frac{1}{\sqrt{x^2+1}}$$

$$y(0) = 1.3$$

$$0.76y(1) + y'(1) = 0.23$$

1.2 Аппроксимация производной

Аппроксимацию производим по следующим формулам:

$$y_i'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O(h^2) \tag{1}$$

$$y_i' = \frac{y_{i+1} - y_{i-1}}{2h} + O(h^2) \tag{2}$$

Существуют также и другие формулы для аппроксимации с точностью $O(h)$. Но для данного метода и требуемой точности были выбранны именно формулы (1) и (2). Посмотрим, что получается, если подставить правые части (1) и (2) на место производных в исходном уравнении (опускаем написание $O(h^2)$).

$$y_i'' + p_i y_i' + q_i y_i = f_i$$

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i\, \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i \tag{3}$$

Получаем общий вид для уравнения:

$$y_{i+1}\left(\frac{1}{h^2}+\frac{p_i}{2h}\right)+y_i\left(-\frac{2}{h^2}+q_i\right)+y_{i-1}\left(\frac{1}{h^2}-\frac{p_i}{2h}\right)=f_i$$

Теперь получим формулу для выражения y_2 через y_1 и y_0.

$$y_2\left(\frac{1}{h^2}+\frac{p_1}{2h}\right)+y_1\left(-\frac{2}{h^2}+q_1\right)+y_0\left(\frac{1}{h^2}-\frac{p_1}{2h}\right)=f_1$$

$$y_2=\left(\frac{1}{h^2}+\frac{p_1}{2h}\right)^{-1}\left(f_1-y_1\left(-\frac{2}{h^2}+q_1\right)-y_0\left(\frac{1}{h^2}-\frac{p_1}{2h}\right)\right)=$$

$$=\left(\frac{1}{2h^2}\right)^{-1}(2+p_1h)^{-1}\left(f_1+y_1\left(\frac{2}{h^2}-q_1\right)+y_0\left(\frac{p_1}{2h}-\frac{1}{h^2}\right)\right)=$$

$$=\frac{2h^2}{2+p_1h}\left(f_1+y_1\frac{1}{h^2}(2-q_1h^2)+y_0\frac{1}{2h^2}(p_1h-2)\right)=$$

$$=\frac{2f_1h^2+y_1\cdot 2(2-q_1h^2)+y_0(p_1h-2)}{2+p_1h} \tag{4}$$

Проделаем то же самое для выражения y_{n-2} через y_{n-1} и y_n.

$$y_n\left(\frac{1}{h^2}+\frac{p_{n-1}}{2h}\right)+y_{n-1}\left(-\frac{2}{h^2}+q_{n-1}\right)+y_{n-2}\left(\frac{1}{h^2}-\frac{p_{n-1}}{2h}\right)=f_{n-1}$$

$$y_{n-2}=\left(\frac{1}{h^2}-\frac{p_{n-1}}{2h}\right)^{-1}\left(f_{n-1}-y_{n-1}\left(-\frac{2}{h^2}+q_{n-1}\right)-y_n\left(\frac{1}{h^2}+\frac{p_{n-1}}{2h}\right)\right)=$$

$$=\left(\frac{1}{2h^2}\right)^{-1}(2-p_{n-1}h)^{-1}\left(f_{n-1}+y_{n-1}\left(\frac{2}{h^2}-q_{n-1}\right)-y_n\left(\frac{p_{n-1}}{2h}+\frac{1}{h^2}\right)\right)=$$

$$=\frac{2h^2}{2-p_{n-1}h}\left(f_{n-1}+y_{n-1}\frac{1}{h^2}(2-q_{n-1}h^2)-y_n\frac{1}{2h^2}(p_{n-1}h+2)\right)=$$

$$=\frac{2f_{n-1}h^2+y_{n-1}\cdot 2(2-q_{n-1}h^2)-y_n(p_{n-1}h+2)}{2-p_{n-1}h} \tag{5}$$

Теперь переходим к работе с уравнениями граничных условий.

Аппроксимация производных для краев будет производиться по следующим формулам:

$$y_i' = \frac{3y_i - 4y_{i-1} + y_{i-2}}{2h} + O(h^2) \tag{6}$$

$$y_i' = \frac{-3y_i + 4y_{i+1} - y_{i+2}}{2h} + O(h^2) \tag{7}$$

Поскольку требуется найти решение с точностью $O(h^2)$, приходится использовать именно формулы (6) и (7). Однако, при подстановке их на место y_0' и y_n' появляется «лишнее значение» – y_2 и y_{n-2} соответственно для уравнений левого и правого концов. Иными словами, в уравнении было две неизвестные, а стало три. Разрешение данной ситуации будет показано ниже при непосредственном вычислении y_n (рассматривается именно правая прогонка).

В уравнении $\alpha_1 y_0 + \alpha_2 y_0' = A$ сделаем подстановку согласно (7).

$$(7) \Rightarrow \quad \alpha_2 y_0' = \alpha_2 \frac{-3y_0 + 4y_1 - y_2}{2h}$$

$$\frac{(2\alpha_1 h - 3\alpha_2)y_0 + 4\alpha_2 y_1 - \alpha_2 y_2}{2h} = A$$

$$(2\alpha_1 h - 3\alpha_2)y_0 + 4\alpha_2 y_1 - \alpha_2 y_2 = 2hA$$

Подставим вместо y_2 правую часть выражения (4)

$$(2\alpha_1 h - 3\alpha_2)y_0 + 4\alpha_2 y_1 - \alpha_2 \frac{2f_1 h^2 + y_1 \cdot 2(2 - q_1 h^2) + y_0(p_1 h - 2)}{2 + p_1 h} = 2hA$$

$$y_0 \underbrace{\left((2\alpha_1 h - 3\alpha_2) - \alpha_2 \frac{p_1 h - 2}{2 + p_1 h}\right)}_{=:\, z} + y_1 \underbrace{\left(4\alpha_2 - \frac{2\alpha_2(2 - q_1 h^2)}{2 + p_1 h}\right)}_{=:\, r} = 2hA + \alpha_2 \frac{2f_1 h^2}{2 + p_1 h}$$

$$y_0 = -\frac{r}{z} y_1 + \frac{2hA + \frac{2f_1 h^2 \alpha_2}{2 + p_1 h}}{z}$$

Введенные новые коэффициенты, обозначенные как r и z, выражаются через известные буквенные коэффициенты, через значения функций $p(x), q(x)$ и $f(x)$ в точке x_1 и шаг h. Все это уже известно нам на данном шаге, а значит, r и z могут быть непосредственно вычислены в программе.

Аналогичная работа проделана с уравнением для правого конца:

$$\beta_1 y_n + \beta_2 y_n' = B$$

$$(6) \Rightarrow \quad \beta_2 y_n' = \beta_2 \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h}$$

$$(2\beta_1 h + 3\beta_2)y_n - 4y_{n-1}\beta_2 + y_{n-2}\beta_2 = 2hB$$

Вместо y_{n-2} подставляем правую часть выражения (5)

$$(2\beta_1 h+3\beta_2)y_n-4y_{n-1}\beta_2+\beta_2\frac{2f_{n-1}h^2+y_{n-1}\cdot 2(2-q_{n-1}h^2)-y_n(p_{n-1}h+2)}{2-p_{n-1}h}=2hB$$

$$y_n\underbrace{\left((2\beta_1 h+3\beta_2)-\beta_2\frac{2+p_{n-1}h}{2-p_{n-1}h}\right)}_{=:\,z}+y_{n-1}\underbrace{\left(-4\beta_2+\frac{2\beta_2(2-q_{n-1}h^2)}{2-p_{n-1}h}\right)}_{=:\,r}=2hB-\beta_2\frac{2f_{n-1}h^2}{2-p_{n-1}h}$$

$$y_n=-\frac{r}{z}y_{n-1}+\frac{2hB-\frac{2f_{n-1}h^2\beta_2}{2-p_{n-1}h}}{z}$$

Запишем полученные для y_0 и y_n выражения в более простом виде: $y_0=\kappa_1 y_1+\nu_1,\ y_n=\kappa_2 y_{n-1}+\nu_2$. Здесь нововведенные коэффициенты заменяют сложные дроби, полученные выше при выводе y_0 и y_n. Сделаем подобную замену и для коэффициентов при y_{i+1}, y_i, y_{i-1} в основном уравнении. И посмотрим, что получается дальше.

1.3 Матрица системы и метод правой прогонки

Теперь является возможным составить систему уравнений, где отсутствуют производные функции $y(x)$. Были ранее получены и описаны уравнения, которые можно представить в таком виде, где для очевидного удобства все коэффициенты переобозначены, чтобы можно было их коротко записать в виде одной буквы.

$$\begin{cases} y_0=\kappa_1 y_1+\nu_1 \\ a_i y_{i+1}-b_i y_i+c_i y_{i-1}=g_i \qquad i=1,...,n-1 \\ y_n=\kappa_2 y_{n-1}+\nu_2 \end{cases} \tag{8}$$

$$a_i=1+\frac{h}{2}p_i,\quad b_i=2-h^2q_i,\quad c_i=1-\frac{h}{2}p_i,\quad g_i=h^2f_i$$

Эта система соответствует следующему матричному выражению с трехдиагональной матрицей:

$$\begin{pmatrix} 1 & -\kappa_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ c_1 & -b_1 & a_1 & 0 & \dots & 0 & 0 & 0 \\ 0 & c_2 & -b_2 & a_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & c_{n-1} & -b_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 & -\kappa_2 & 1 \end{pmatrix}\cdot\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}=\begin{pmatrix} \nu_1 \\ g_1 \\ g_2 \\ \vdots \\ g_{n-1} \\ \nu_2 \end{pmatrix}$$

Будем искать решение задачи в том же виде, в котором заданы краевые условия. В методе правой прогонки используется вид, соответствующий уравнению для левого края:

$$y_i=u_i y_{i+1}+v_i,\quad i=1,...,n-1$$

где u_i, v_i – какие-то коэффициенты, которые предстоит найти.

Из среднего уравнения системы (8) для номера i исключаем y_{i-1}:

$$y_{i-1} = u_{i-1}y_i + v_{i-1}$$

Подставим это выражение в среднее уравнение (8)

$$a_i y_{i+1} - b_i y_i + c_i(u_{i-1}y_i + v_{i-1}) = g_i$$

$$a_i y_{i+1} - y_i(b_i - c_i u_{i-1}) = g_i - c_i v_{i-1}$$

$$y_i = \underbrace{\frac{a_i}{b_i - c_i u_{i-1}}}_{u_i} y_{i+1} + \underbrace{\frac{c_i v_{i-1} - g_i}{b_i - c_i u_{i-1}}}_{v_i}$$

Получаем рекуррентные соотношения для определения **прогоночных коэффициентов** u_i и v_i

$$u_i = \frac{a_i}{b_i - c_i u_{i-1}}, \quad v_i = \frac{c_i v_{i-1} - g_i}{b_i - c_i u_{i-1}}, \quad i = 1, ..., n-1 \tag{9}$$

u_0 и v_0 находим из краевого условия, то есть, $u_0 = \kappa_1,\ v_0 = \nu_1$

Значение y_n, необходимое для начала счета по формулам рекуррентных соотношений, получаем из краевого условия на правом конце и уравнения $y_{n-1} = u_{n-1}y_n + v_{n-1}$, то есть из следующей системы:

$$\begin{cases} y_{n-1} - u_{n-1}y_n = v_{n-1} \\ -\kappa_2 y_{n-1} + y_n \ = \nu_2 \end{cases}$$

Откуда y_n выражается следующим образом:

$$y_n = \frac{-\kappa_2 v_{n-1} - \nu_2}{\kappa_2 u_{n-1} - 1} \tag{10}$$

Теперь есть все, чтобы перейти непосредственно к вычислению значений функции в узлах сетки при помощи компьютерной программы.

1.4 Код программы

Программирование мной ведется на языке Fortran, поэтому вычислительные выражения описаны именно на этом языке. Весь код программы указан в конце данного раздела.

Для начала нас интересует значение n количества промежутков, на которые будет разбит весь отрезок $[a, b]$. Это значение запрашивается у пользователя компьютера во время выполнения работы программы.

Функции $p(x), q(x), f(x)$, а также коэффициенты для уравнений граничных условий выписаны отдельно в модуле. В программе функции $p(x), q(x), f(x)$ функциями (в понятиях языка Fortran) и являются (их имена Pfun, Qfun, Ffun), а в коде основной программы можно задавать значения этих функций в точках, используя запись через название функции, указав под x значение абсциссы.

Зная n, можно вычислить шаг h, а также создать рабочие массивы со значениями функции y в узлах, массив самих точек x_i, и массив значений функций p_i, q_i, f_i.

forall (i=0:n) X(i)=a+i*h
forall (i=0:n) P(i)=Pfun(X(i)), и аналогично для Q(i) и F(i)

Рассмотрим блок следующих формул, встречающийся в выводе уравнений для y_0:

$$y_0 \underbrace{\left((2\alpha_1 h - 3\alpha_2) - \alpha_2 \frac{p_1 h - 2}{2 + p_1 h}\right)}_{=:\, z} + y_1 \underbrace{\left(4\alpha_2 - \frac{2\alpha_2(2 - q_1 h^2)}{2 + p_1 h}\right)}_{=:\, r} = 2hA + \alpha_2 \frac{2 f_1 h^2}{2 + p_1 h}$$

$$y_0 = -\frac{r}{z} y_1 + \frac{2hA + \frac{2 f_1 h^2 \alpha_2}{2 + p_1 h}}{z}$$

$$y_0 = \kappa_1 y_1 + \nu_1$$

Эти формулы именно так и пишутся в программе. Сначала вычисляются значения r и z, так как для них известны все буквенные коэффициенты и значения всех функций в точке x_1.

При сравнении этих двух выражений для y_0 друг с другом отчетливо видно, чему равны коэффициенты κ_1 и ν_1, ввиду одинаковой формы их записи относительно y_0 и y_1. Как видно, при уже вычисленных коэффициентах r и z и при известных остальных буквенных коэффициентах, можно вычислить κ_1 и ν_1.

В выводе формул для уравнения, задающего краевые условия для правого конца, были получены аналогичные формулы с r и z и с κ_2 и ν_2 соответственно. Путь вычисления коэффициентов κ_2 и ν_2 по формулам для y_n совершенно аналогичен.

Теперь создаются векторы U(0:n-1) и V(0:n-1) и прямоугольная матрица размера $((n-1) \times 4)$ для заполнения ее значениями коэффициентов

$$a_i = 1 + \frac{h}{2} p_i, \quad b_i = 2 - h^2 q_i, \quad c_i = 1 - \frac{h}{2} p_i, \quad g_i = h^2 f_i$$

для вычисления которых также все известно: как шаг h, так и значения функций в точках x_i.

Так как на данном шаге известны все коэффициенты $a_i, b_i, c_i, g_i, \kappa_1, \nu_1$, уже можно приступить к получению векторов U(0:n-1) и V(0:n-1), значения которых выражаются по формулам (9) при $u_0 = \kappa_1,\ v_0 = \nu_1$.

Затем по формуле (10) вычисляется значение y_n, которое нужно для «старта» прогонки.

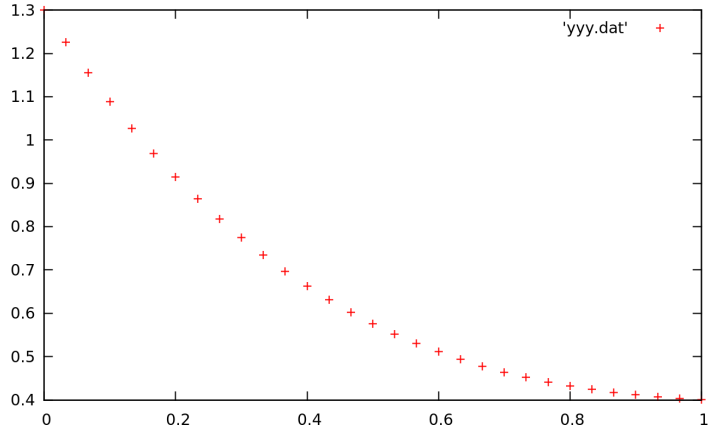
Наконец, используя введенное рекуррентное соотношение $y_i = u_i y_{i+1} + v_i$ в цикле

```
do i=n-1,0,-1
Y(i)=U(i)*Y(i+1)+V(i)
enddo
```

производится вычисление всех значений Y(0:n), что и является конечным продуктом решения задачи данным методом.

Результаты вычислений для $n = 30$

i	x_i	y_i	i	x_i	y_i
0	0.000000	1.3000000000000000	16	0.533333	0.55234754587956436
1	0.033333	1.2254640054754096	17	0.566666	0.53074317083503297
2	0.066666	1.1552732275217867	18	0.599999	0.51126163495351562
3	0.100000	1.0893000105379336	19	0.633333	0.49377430443210080
4	0.133333	1.0274125357567254	20	0.666666	0.47815611156509041
5	0.166666	0.96947540912564656	21	0.699999	0.46428578909338125
6	0.200000	0.91535025558068184	22	0.733333	0.45204606973631334
7	0.233333	0.86489631452695559	23	0.766666	0.44132385189012030
8	0.266666	0.81797103140093597	24	0.800000	0.43201033272886757
9	0.299999	0.77443064034672737	25	0.833333	0.42400111015206809
10	0.333333	0.73413073328932799	26	0.866666	0.41719625519060055
11	0.366666	0.69692681102210852	27	0.900000	0.41150035661218093
12	0.400000	0.66267481233157921	28	0.933333	0.40682253956280495
13	0.433333	0.63123161764433477	29	0.966666	0.40307646014472243
14	0.466666	0.60245552418201964	30	1.000000	0.40018027786803473
15	0.500000	0.57620669013316383			



1.5 Проверка

Полученные программой значения подверглись разным вариантам проверки. Один из них выглядит следующим образом:

Была рассмотрена функция $g(x) = x^3 + 3x^2 + x + 3$, про которую предполагается, что она является решением дифференциального уравнения. Функции $p(x)$ и $q(x)$ взяты такие же, какие были даны в условии моей задачи. Коэффициенты $\alpha_1, \alpha_2, \beta_1, \beta_2$ – те же. А вот функция $f(x)$ и коэффициенты A и B определяются новые:

$$g(x) = x^3 + 3x^2 + x + 3 = (x + 3)(x^2 + 1)$$

$$g'(x) = 3x^2 + 6x + 1$$

$$g''(x) = 6x + 6$$

$$g''(x) + p(x)g'(x) + q(x)g(x) = f(x)$$

$$(6x + 6) + (x + \frac{3}{2})(3x^2 + 6x + 1) + \frac{x+1}{x+3}(x + 3)(x^2 + 1) =$$

$$= 6x + 6 + 3x^3 + 6x^2 + x + 4.5x^2 + 9x + 1.5 + x^3 + x^2 + x + 1 =$$

$$= 4x^3 + 11.5x^2 + 17x + 8.5 =: f(x)$$

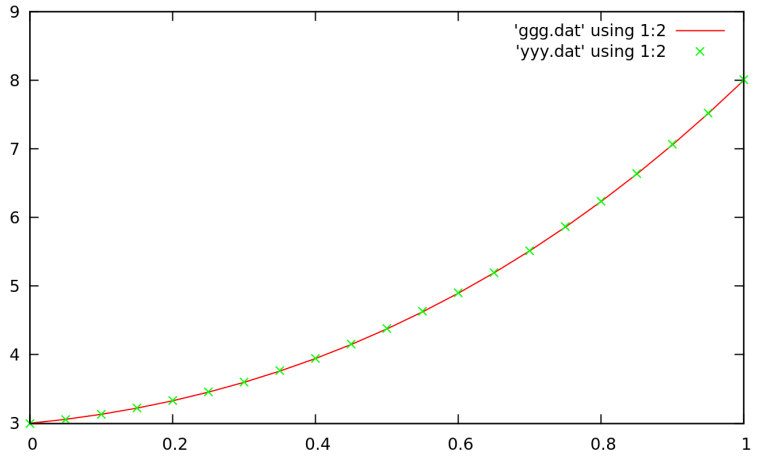
$$g(0) = 3 =: A$$

$$0.76g(1) + g'(1) = 0.76(1 + 3 + 1 + 3) + (3 + 6 + 1) = 16.08 =: B$$

Таким образом, новые условия задачи имеют следующий вид:

$$\begin{cases} y''(x) + (x + \frac{3}{2})\, y'(x) + \frac{x+1}{x+3}\, y(x) = 4x^3 + 11.5x^2 + 17x + 8.5 \\ y(0) = 3 \\ 0.76y(1) + y'(1) = 16.08 \end{cases}$$

Набор значений y_i $(i = 0, ..., n)$, получаемый при решении задачи с новыми $f(x), A, B$, действительно совпадает со значениями функции $g(x)$ в точках x_i с заявленной $O(h^2)$ точностью для разных h, что наглядно показано на графике (при $h = 20$). ggg.dat – значения функции $g(x)$, yyy.dat – значения y_i $(i = 0, ..., n)$, получаемые программой.



```
program diffur
use funs
implicit none
integer :: n, i
real(8) :: h, kap1, nu1, kap2, nu2, r, z
real(8), allocatable :: Y(:), X(:), P(:), Q(:), F(:)
real(8), allocatable :: kf(:,:), U(:), V(:)
write(*,*) 'input N'; read(*,*) n; h=(b-a)/n
allocate (Y(0:n), X(0:n), P(0:n), Q(0:n), F(0:n))
forall (i=0:n) X(i)=a+i*h
forall (i=0:n)
    P(i)=Pfun(X(i))
    Q(i)=Qfun(X(i))
    F(i)=Ffun(X(i))
end forall
!---------------------------------------
r=(4.0d0-2.0d0*(2.0d0-Q(1)*h**2)/(2.0d0+P(1)*h))*alp2
z=2.0d0*alp1*h-3.0d0*alp2-alp2*(P(1)*h-2.0d0)/(2.0d0+P(1)*h)
kap1=-r/z
nu1=(2.0d0*h*Abig+2.0d0*alp2*F(1)*h**2/(2.0d0+P(1)*h))/z

r=(-4.0d0+2.0d0*(2.0d0-Q(n-1)*h**2)/(2.0d0-P(n-1)*h))*bet2
z=2.0d0*bet1*h+3.0d0*bet2-bet2*(2.0d0+P(n-1)*h)/(2.0d0-P(n-1)*h)
kap2=-r/z
nu2=(2.0d0*h*Bbig-2.0d0*bet2*F(n-1)*h**2/(2.0d0-P(n-1)*h))/z
!---------------------------------------
allocate (kf(4,1:n-1), U(0:n-1), V(0:n-1))
forall (i=1:n-1)
    kf(1,i)=1.0d0+h/2.0d0*P(i)
    kf(2,i)=2.0d0-h**2*Q(i)
    kf(3,i)=1.0d0-h/2.0d0*P(i)
    kf(4,i)=h**2*F(i)
end forall
U(0)=kap1; V(0)=nu1
do i=1,n-1
    U(i)=kf(1,i)/(kf(2,i)-kf(3,i)*U(i-1))
    V(i)=(kf(3,i)*V(i-1)-kf(4,i))/(kf(2,i)-kf(3,i)*U(i-1))
enddo
Y(n)=(-kap2*V(n-1)-nu2)/(kap2*U(n-1)-1.0d0)
do i=n-1,0,-1; Y(i)=U(i)*Y(i+1)+V(i); enddo
open(777,file='yyy.dat')
    do i=0,n
        write(777,*) X(i), Y(i)
    enddo
close(777)
end program diffur
```

```
module funs
implicit none
real(8) :: a=0.0d0, b=1.0d0
real(8) :: alp1=1.0d0, alp2=0.0d0, Abig=1.3d0
real(8) :: bet1=0.76d0, bet2=1.0d0, Bbig=0.23d0

contains

pure function Ffun(x) result(f)
implicit none
real(8), intent(in) :: x
real(8) :: f
!---------------------------------------
f=1/sqrt(x**2+1.0d0)
end function Ffun

pure function Pfun(x) result(p)
implicit none
real(8), intent(in) :: x
real(8) :: p
!---------------------------------------
p=x+1.5d0
end function Pfun

pure function Qfun(x) result(q)
implicit none
real(8), intent(in) :: x
real(8) :: q
!---------------------------------------
q=(x+1.0d0)/(x+3.0d0)
end function Qfun

end module funs
```