БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра информатики

Факультет НиДО

Специальность ИиТП

Контрольная работа № 1

по дисциплине «Методы численного анализа»

Выполнил студент: Дегтярев А.А.

группа 393551

Зачетная книжка № 902021-26

Минск 2017

### ИПР №1. Решение краевых задач методом разностных аппроксимаций.

|  |
| --- |
| ЗАДАНИЕ. Составить разностную схему и получить численное решение краевой задачи с точностью 10-3 |

Нам дано дифференциально уравнение второго порядка вида

Разобьем отрезок [a,b] на n одинаковых частей с шагом h = (b-a)/n

Точками A=x0<x1..<xn=b заменяем



где 

И для любого внутреннего узла получим уравнение



из n+1 уравнений мы получим систему с неизвестным yk, решив систему – получим приближенное решение краевой задачи

|  |
| --- |
| function ipr1()      syms x      xlb = -1;      xub = 1;      n = 10;        b = cos(degtorad(26));      c = sin(degtorad(26));      q=-(1+b\*x\*x)/c;      f=-1/c;        for n = 10:10:50          Y = sym('y',[1,n]);          xn = zeros(1,n);            sys = sym('f',[1,n]);          dx=((xub-xlb)/n);            sys(1)=Y(1)+1;          xn(1) = xlb;          sys(n)=Y(n);          xn(n) = xub;            for i = 2:(n-1)              xn(i) = xlb+dx\*i;              pxn = subs(p,'x',xn(i));              qxn = subs(q,'x',xn(i));              fxn = subs(f,'x',xn(i));              sys(i) = (Y(i+1)-2\*Y(i)+Y(i-1))/(dx\*dx) + pxn\*((Y(i+1)-Y(i))/(2\*dx)) - qxn\*Y(i) - fxn;              sys(i) = sys(i);          end            sysRes = solve(sys,Y);            for i = 1:n              idx{1} = char(Y(i));              Y(i)=(sysRes.(idx{1}));          end            plot(xn,Y); hold on      end end |



### ИПР №2. Решение задачи теплопроводности методом разностных аппроксимаций

|  |
| --- |
| ЗАДАНИЕ НА РАБОТУ №2. Промоделировать нестационарные процессы теплопроводности в зависимости от входных данных задачи - коэффициента теплопроводности k(x) и начальной температуры :    1. Найти приближенное решение задачи с шагами  и *h*=0.01, используя   **явную разностную схему**. Построить графики решений при значениях  *t*= 0.5, 20, 200.  2. Используя результаты экспериментально определить момент времени *t*, при котором происходит установление процесса (визуально).  3. Исследовать, как влияет начальная температура на процесс установления, взяв другие функции  (согласованные с граничными условиями). |

Задача представляет собой одномерный случай диффузионного процесса. Так как f(x) и (k(x) не зависят от температуры, уравнение считается линейным.

Начальные и граничные условия задают начальный нагрев центральной области стержня, и поддержание постоянно температуры; Для решения применяют как явные, так и неявные разностные схемы; Для составления разностной схемы использовался 4-точечный метод в виде перевернутой буквы Т; Покроем область сеткой, образованной прямыми и будем определять значения решения в узлах сетки;

|  |
| --- |
| function ipr2()      syms x      l = 1;      xlb = 0;      xub = l;        syms t      T = 0.2;      tlb = 0;      tub = T;        n = 13;      U = sym('u',[n,n]);        xn = zeros(1,n);      tn = zeros(1,n);        a = 0      b = 1      UA = 0      UB = 1        fx = 3\*log(x);      gx = (1-exp(-t))\*fx;      kx = 1/x;      sys = sym('f',[n,n]);      dx=((xub-xlb)/n);      dt=((tub-tlb)/n);        for j = 1:n          sys(1,j)=U(1,j)-UA;          sys(n,j)=U(n,j)-UB;      end        for i = 1:n          sys(i,1)=U(i,1)-a;          sys(i,n)=U(i,n)-b;      end        for i = 2:(n-1)          xn(i) = xlb+dx\*i;          for j = 2:(n-1)              tn(j) = tlb+dt\*j;              sys(i,j) = ((U(i,j+1)-U(i,j))/dt) - subs(kx,'x',xn(i)) \* ((U(i+1,j)-2\*U(i,j)+U(i-1,j))/(dx\*dx)) -subs(subs(gx,'x',xn(i)),'t',tn(j));          end      end        flatSys = sym('f',[1,n\*n]);      flatU = sym('u',[1,n\*n]);      for i = 1:n          for j = 1:n              flatIDX = i\*n+j-n;              flatU(flatIDX) = U(i,j);              flatSys(flatIDX)= sys(i,j);          end      end        flatSysRes = solve(flatSys,flatU);        sysRes = zeros(n,n);      for i = 1:n          for j = 1:n              flatIDX = i\*n+j-n;              idx{1} = char(flatU(flatIDX));              sysRes(i,j) = (flatSysRes.(idx{1}))/(n\*n);          end      end         mesh(tn,xn,sysRes);hold on  end |



### КОНТРОЛЬНАЯ РАБОТА №1

### Решение краевых задач. Методы коллокаций, наименьших квадратов и Галеркина.

|  |
| --- |
| ЗАДАНИЕ. Методами коллокаций, интегральным и дискретным методами наименьших квадратов и Галеркина получить численное решение краевой задачи   Исходные данные:  ,где k номер варианта.  Базисную систему выбрать в виде: |

Нам дано дифференциально уравнение второго порядка

Описанные здесь методы, являются проекционными;

Базисная система указана в начальных условиях

Поскольку функции линейно независимы

Строим приближенное решение с использованием базисной системы:

Невязка - функция составленная из разности левой и правой части уравнения, образующаяся при подстановке y2(x) вместо Y(x). Которая характеризует степень отклонения функции y2(x) от точного решения; Если при некоторых коэффициентах невязка будет равна нулю, то функция y2 – совпадает с точным решением краевой задачи;

L – линейный оператор задачи.

Все изложенные ниже методы используются для минимизации невязки:

**Метод коллокаций** – на отрезке [a,b] выбираются точки которые называются точками коллокаций. Они последовательно подставляются в невязку, считая, что невязка должны быть равна нулю в точках коллокации. В итоге получим систему уравнений для определения коэффициентов n-n. Решая систему – найдем приближенное решение yn(x).

Выберем n точек коллокации на отрезке [a,b], подставив их в e(x,a..a)=0

Получим систему уравнений из n уравнений вида: i = 1..n

Где

Решая ее получим значения коэффициентов a, и подставив в уравнение

получим приближенное решение.

**Метод наименьших квадратов(непрерывный вариант)**

В этом случае неизвестные коэффициенты a1..an должны обеспечивать минимум интеграла от квадрата невязки.

Получим систему уравнений из n уравнений вида:

где - скалярное произведение

**Метод наименьших квадратов(дискретныйвариант)**

В этом случае неизвестные коэффициенты a1..an должны обеспечивать минимум суммы квадратов значения невязки.

Получим систему уравнений из n уравнений вида:

где - скалярное произведение

В случае когда n=m результаты полученные мнк и методом колокаций совпадают.

**Метод Галеркина**

Неизвестные коэффициенты находятся из условия ортогональности функций базисной системы к невязке

Система примет вид:

где - скалярное произведение

### КОНТРОЛЬНАЯ РАБОТА №2

### Метод сеток решения задачи Дирихле для уравнения Пуассона.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Задание. Решить двухмерное стационарное уравнение Пуассона    Область Ω представляет собой четырехугольник (А)    Участки границы Гi обозначим номерами 1,2,3,4. На границах Гi заданы следующие условия:    g(x,y) = 1+0.5cos(x+y)  f(x,y) = exp(-(y-x)2   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | 1 | 2 | 3 | 4 | | q | 0 | 0 | 1 | 1 | | a | 1 | 1 | -0.5 | 0 | | b | -2 | -2 | -0.8 | -0.2 |   Решить уравнение, используя последовательность сгущающихся сеток (три раза), проанализировать сходимость метода и погрешность решения. |

Задача Дирихле состоит в следующем. Требуется найти непрерывную на  функцию u (х, у), удовлетво­ряющую на открытом квадрате D уравнению Пуассона

|  |  |  |
| --- | --- | --- |
|  | = f(x, y) |  |

и обращающуюся на границе квадрата в нуль

аналогично задаче о теплопроводности; построим сетку x=ih и y=jh.

Пусть область D покрыта сеткой, пересечения – узлы сетки; Аппроксимируем область D сетчатой областью состоящей из всех квадратов целиком лежащих в области D; Контур аппроксимируется контуром из отрезков сетки; Производные заменяются конечными разностями; Уравнение заменяется уравнением в конечных разностях, далее для каждого узла сетки составляется такое уравнение; Используется шаблон «крест». Если точка соседняя с точкой контура, то в правой части уравнения слагаемые (известные) заменяются; Далее получим неоднородную систему уравнений, решив которую получим приблизительное решение задачи.

|  |
| --- |
| function kr2()   syms x y;   xlb = 0;   xub = 1;   ylb = 0;   yub = 1;   fx = x\*x+y\*y;%exp(-(y-x)^2);   gx = 1+x+y;%1+0.5\*cos(x+y);   n0 = 5;     for k = n0:10       n = k;       U = sym('u',[n,n]);         xn = zeros(1,n);       yn = zeros(1,n);       sys = sym('f',[n,n]);       dx=((xub-xlb)/n);       dy=((yub-ylb)/n);    %    Need to set actual border values       for i = 1:n          sys(i,1)=U(i,1);          sys(i,n)=U(i,n);       end         for j = 1:n          sys(1,j)=U(1,j);          sys(n,j)=U(n,j);       end    %   Need to check for border nodes to skip calc on them       for i = 2:n-1          xn(i) = xlb+dx\*i;          for j = 2:n-1              yn(j) = ylb+dy\*j;              gnx = subs(subs(gx,x,xn(i)),y,yn(j));              fnx = subs(subs(fx,x,xn(i)),y,yn(j));              sys(i,j) = (-fnx+gnx\*(U(i+1,j)+U(i-1,j)-2\*U(i,j))/(dx\*dx)+(U(i,j+1)+U(i,j-1)-2\*U(i,j))/(dy\*dy));          end       end        flatSys = sym('f',[1,n\*n]);      flatU = sym('u',[1,n\*n]);      for i = 1:n          for j = 1:n              flatIDX = i\*n+j-n;              flatU(flatIDX) = U(i,j);              flatSys(flatIDX)= sys(i,j);          end      end        flatSysRes = solve(flatSys,flatU);        sysRes = zeros(n,n);      for i = 1:n          for j = 1:n              flatIDX = i\*n+j-n;              idx{1} = char(flatU(flatIDX));              sysRes(i,j) = (flatSysRes.(idx{1}))/(n\*n);          end      end      nxn = xn(2:(n-1));      nyn = yn(2:(n-1));      nsysres = sysRes(2:(n-1),2:(n-1));      mesh(nxn,nyn,nsysres);hold on  end  end    function res = kr2\_test\_g(x,y)      %not yet implemented  end |