Описание алгоритма решения уравнения Пуассона в прямоугольной области

1. Прямой ход рекурсии

Входные данные

X1, X2 – границы прямоугольной области по оси X

Y1, Y2 – границы прямоугольной области по оси Y

NX – количество разбиений по оси X, должно быть $2^n + 1$ точек (в некоторых вариантах допустимо $7\times2^n + 1$ точек)

NY – количество разбиений по оси Y, должно быть $2^n + 1$ точек (в некоторых вариантах допустимо $7\times2^n + 1$ точек)

$$h_x = \frac{X2 - X1}{NX}$$

$$h_y = \frac{Y2 - Y1}{NY}$$

Массивы граничных значений функции Uleft(NY), Uright(NY), Ubottom(NX), Utop(NX)

Массив значений правой части в узлах сетки – сеточная функция! (У Голдырева было сделано криво – не было входного массива!) F(NX, NY)

Внимание! Именно так, нужно значение правой части в точках, сколь угодно близких к граничным, нельзя ограничиваться лишь внутренними точками области.

Этап 1.1 Деление на четную и нечетную части вдоль направления оси X

Это промежуточный шаг алгоритма.

На этом этапе получаем некоторые промежуточные массивы.

Сдвиг по координате, $x = \frac{X1 - X2}{2} + X - X1$

Вместо массива правых частей получаются два массива размера $2^{n-1} + 1$ точек в направлении x.

Расчетные формулы

Внимание! При реализации этапа разумна индексация массива от (1 – NX)/2 до (NX – 1)/2 включительно!

При реализации этапа разумна индексация массива от (1 – NY)/2 до (NY – 1)/2 включительно!

Рассмотрим постановку эллиптической задачи с граничными условиями первого рода

Пусть уравнение Пуассона

$$\Delta u = f(x, y) \tag{1}$$

решается в области $\Omega: \{-1 \le x \le 1, -1 \le y \le 1\}$ с условиями

После сдвига по x первая переменная лежит именно в этих пределах, сдвиг по y описан ниже.

$$u(-1, y) = f_1(y)$$
$$u(1, y) = f_2(y)$$
$$u(x, -1) = f_3(x)$$
$$u(x, 1) = f_4(x).$$

Кроме этой задачи, для уравнения Пуассона

$$\Delta v = g(x, y) = f(-x, y)$$

в той же области $\Omega: \{-1 \le x \le 1, -1 \le y \le 1\}$ поставлена задача с условиями

$$v(-1, y) = f_2(y)$$
$$v(1, y) = f_1(y)$$
$$v(x, -1) = f_3(-x)$$

$v(x,1) = f_4(-x).$

Данная задача получена из Задачи 1 заменой x на $-x$.

Верно равенство

$v(x,y) = u(-x,y)$

Введем две новые функции

$$w_0 = \frac{1}{2}(u+v), \quad w_1 = \frac{1}{2}(u-v).$$

Применим к этим функциям оператор Лапласа, получим

$$\Delta w_0 = \frac{1}{2}\Delta(u+v) = \frac{1}{2}(\Delta u + \Delta v) = \frac{f(x,y)+f(-x,y)}{2} = \varphi(x,y).$$

На границах области для функции выполняются условия

$$w_0(-1,y) = \frac{1}{2}(f_1(y)+f_2(y)) = \varphi_0(y),$$

$$w_0(1,y) = \frac{1}{2}(f_1(y)+f_2(y)) = \varphi_0(y),$$

$$w_0(x,-1) = \frac{1}{2}(f_3(x)+f_3(-x)) = \varphi_3(x),$$

$$w_0(x,1) = \frac{1}{2}(f_4(x)+f_4(-x)) = \varphi_4(x),$$

В силу того, что функция четная, выполнено $\dfrac{\partial w_0(0,y)}{\partial x} = 0$.

Аналогично

$$\Delta w_1 = \frac{1}{2}\Delta(u-v) = \frac{1}{2}(\Delta u - \Delta v) = \frac{f(x,y)-f(-x,y)}{2} = \psi(x,y).$$

На границах области для функции выполняются условия

$$w_1(-1,y) = \frac{1}{2}(f_1(y)-f_2(y)) = \psi_{-1}(y),$$

$$w_1(1,y) = \frac{1}{2}(-f_1(y)+f_2(y)) = -\psi_{-1}(y) = \psi_1(y),$$

$$w_1(x,-1) = \frac{1}{2}(f_3(x)-f_3(-x)) = \psi_3(x),$$

$$w_1(x,1) = \frac{1}{2}(f_4(x)-f_4(-x)) = \psi_4(x),$$

В силу того, что функция нечетная, выполнено $w_1(0,y) = 0$.

Каждую из этих двух функций можно искать на той же сетке (с той же точностью ε, определяемой аппроксимацией, или с точностью ε/2), что и исходную, но в области $\Omega_1 : \{0 \le x \le 1, -1 \le y \le 1\}$

Далее, чтобы не путаться, у новых выходных данных появятся числовые обозначения. 0 на первом месте – четная по x функция, 1 –нечетная по x функция. 0 на втором месте – четная по y функция, 1 на втором месте – нечетная по y функция.

Массив F0 ([nx/2]+1, NY) – правые части для «четного» уравнения (см. предыдущий документ с расчетными формулами) F1 ([nx/2]+1, NY) – правые части для «нечетного» уравнения.

Появляются 4 новых граничных массива Ubottom01([nx/2]+1), Utop00([nx/2]+1), Ubottom11([nx/2]+1), Utop10([nx/2]+1)

В массивах *0* лежат четные (по переменной x) части, в массивах *1* – нечетные

Модифицируются значения граничных массивов на правой и левой границах.

Старые нам необходимо сохранить, следовательно, появятся Uleft1(NY), Uright0(NY), содержащие «четные» и «нечетные» части (по переменной x).

Появляется еще один вспомогательный массив Uright1(NY), соответствующий правому краевому условию для нечетной задачи. Так как решение суть нечетная функция, то массив содержит одни нули.

Этап 1.2 Построение дополнительных граничных массивов с использованием оператора интерполяции.

Чтобы завершить первый этап рекурсии нам не хватит данных. Для восполнения используем решение уравнения Пуассона на грубой 9-точечной сетке.

на девятиточечном шаблоне

$$u(0,0) = \tfrac{\sqrt{2}-1}{4}(u(-1,0) + u(1,0) + u(0,-1) + u(0,1) + f(0,0)) +$$

$$+\tfrac{2-\sqrt{2}}{4}(u(-1,-1) + u(1,-1) + u(-1,1) + u(1,1) + f(0,0))$$

Формула неверна – не вяжется сумма коэффициентов и размерность!

При x=0 строим интерполяцию полиномом 2 порядка по переменной *y*. Значения функции в узлах сетки помещаем в массив Uleft0(NY).

При y=0 строим интерполяцию полиномом 2 порядка по переменной *x*. Значения функции в узлах сетки помещаем в массив Ubottom0*(NX) (*= 0 и 1 соответственно).

Этап 1.3 Деление на четную и нечетную части вдоль направления оси Y

Это промежуточный шаг алгоритма, напоминающий этап 1.1

На этом этапе получаем некоторые промежуточные массивы, но для данного этапа рекурсии они уже играют роль выходных параметров.

Сдвиг по координате, $y = \frac{Y1-Y2}{2} + Y - Y1$

Расчетные формулы

Заметим, что каждую из получившихся задач можно разбить еще на две подзадачи в области $\Omega_2 : \{0 \le x \le 1,\ 0 \le y \le 1\}$. На этом этапе используется уже разделение на четную и нечетную части по направлению *y*.

Пусть уравнение Пуассона

$$\Delta w_0 = \varphi(x,y) \qquad (3)$$

решается в области $\Omega_{1/2} : \{0 \le x \le 1,\ -1 \le y \le 1\}$ с условиями

$$\frac{\partial}{\partial x} w_0(0,y) = 0,$$

$$w_0(1,y) = \varphi_1(y)$$

$$w_0(x,-1) = \varphi_3(x),$$

$$w_0(x,1) = \varphi_4(x).$$

Кроме этой задачи, для уравнения Пуассона

$$\Delta w_{0,-1} = \varphi(x,-y),$$

в той же области $\Omega_{1/2} : \{0 \le x \le 1,\ -1 \le y \le 1\}$ поставлена задача с условиями

$\frac{\partial}{\partial x} w_{0,-1}(0,y) = 0,$ Вместо этого условия второго рода используем условие первого рода, полученное с помощью интерполяции на Этапе 1.1

$$w_{0,-1}(1,y) = \varphi_1(-y),$$

$$w_{0,-1}(x,-1)=\varphi_4(x)$$
$$w_{0,-1}(x,1)=\varphi_4(x).$$

Данная задача получена из Задачи 3 заменой *y* на –*y*.

Снова введем в рассмотрение две новые функции

$$w_{00}=\frac{1}{2}(w_0+w_{0,-1}),\quad w_{01}=\frac{1}{2}(w_0-w_{0,-1}).$$

Так как первая из этих функция четная по *y*, то $\frac{\partial w_{00}}{\partial y}(x,0)=0$. Для второй функции верно $w_{01}(x,0)=0$. Для каждой из этих функций может быть поставлена задача в области, например, $\Omega_{1/4}:\{0\le x\le 1,0\le y\le 1\}$.

Аналогично задача для функции w_1 может быть разделена на задачи для двух функций $w_{01}=\frac{1}{2}(w_1+w_{1,-1})$, $w_{11}=\frac{1}{2}(w_1-w_{1,-1})$. $\Omega_{1/4}:\{0\le x\le 1,0\le y\le 1\}$.

Каждая из этих задач может независимо решаться на своем исполнителе.

Сначала обрабатываем массивы F0 ([nx/2]+1, NY) – правые части для «четного» уравнения (см. предыдущий документ с расчетными формулами) F1 ([nx/2]+1, NY) – правые части для «нечетного» уравнения. Каждый из них делится на два. F00 ([nx/2]+1, [ny/2]+1) – элементы массива F0, преобразованные для четного по y решения, F01 ([nx/2]+1, [ny/2]+1) – элементы массива F0, преобразованные для нечетного по y решения, . F10 ([nx/2]+1, [ny/2]+1) – элементы массива F1, преобразованные для четного по y решения, F11 ([nx/2]+1, [ny/2]+1) – элементы массива F1, преобразованные для нечетного по y решения.

Появляются 4 новых граничных массива Ubottom01([nx/2]+1), Utop00([nx/2]+1), Ubottom11([nx/2]+1), Utop10([nx/2]+1)

В массивах **0 лежат четные (по переменной x) части, в массивах **1 - нечетные

Модифицируются значения граничных массивов на правой и левой границах.

Мы получим четыре задачи, полностью аналогичные исходной, но вчетверо меньшей размерности

Выходные параметры

X1LM, X2LM – границы прямоугольной области по оси X

Y1LM, Y2LM – границы прямоугольной области по оси Y

NX_I – количество разбиений по оси X, должно быть $2^{n-1}+1$ точек (в некоторых вариантах допустимо $7\times2^{n-1}+1$ точек)

NY_1 – количество разбиений по оси Y, должно быть $2^{n-1}+1$ точек (в некоторых вариантах допустимо $7\times2^{n-1}+1$ точек)

Значения шагов hx, hy не меняются в ходе расчетов.

Массивы граничных значений функции UleftLM(NY_I), UrightLM(NY_I), UbottomLM(NX_I), UtopLM(NX_I)

Массив значений правой части в узлах сетки – сеточная функция! FLM(NX_I, NY_I)

LM принимают значения 00, 01, 10, 11

Если значения NX_I NY_1 не равны 3 или 7, то для каждой из 4 подзадач снова вызывается тот же алгоритм (уже 4 раза с меньшим объемом входных данных для каждого)

Если значения NX_I NY_1 равны 3 или 7, то для каждой из 4 подзадач вызывается процедура решения систем линейных алгебраических уравнений.

Можно попробовать использовать прямой метод (Гаусса) решения СЛАУ.

Выходными параметрами для процедуры решения является массив решений

ULM (NX_I, NY_I)

Обратный ход – восстановление решения

Входные данные алгоритма U00 (NX_I, NY_I), U01 (NX_I, NY_I), U10 (NX_I, NY_I), U11 (NX_I, NY_I), F(NX,NY), tau, NITER, epsITER

На выходе массив решения U(NX, NY)

Объединяем значения в соответствии с формулами

U*0(–k, j) = U00 (k, j) – U10 (k, j)

U*0(k, j) = U00 (k, j) + U10 (k, j)

U*1(–k, j) = U01 (k, j) – U11 (k, j)

U*1(k, j) = U01 (k, j) + U11 (k, j)

U(k,-j)=U*0(k,j) – U*1(k,j)

U(k,j)=U*0(k,j) + U*1(k,j)

Счетчик итераций Niter = 0

Считается вспомогательный массив невязок во «внутренних» точках области

R(k,j)=F(k,j) – AU(k,j)

Здесь A – сеточный оператор Лапласа

Если счетчик итераций Niter не превосходит значения NITER, то считается норма C вектора невязки. Если она меньше epsITER, то из функции выходим, возвращаем массив U

В противном случае уточняем значение U(k,j)=U(k,j) + tau*R(k,j)

Если счетчик итераций Niter достиг значения NITER, то из функции выходим, возвращаем массив U

Niter += 1 и переходим к этапу вычисления невязки для следующей итерации

Примечание. Пока итерационный параметр tau зададим, далее при прямом ходе он будет вычисляться для каждого этапа.

Этими итерациями гасим неточности, порождаемые задачей интерполяции и решением вспомогательной задачи на грубой сетке.