ChatGPT

Теоретическая часть

Общая задача. Искомая задача – стационарное уравнение теплопроводности в прямоугольнике $0<x<a,\;0<y<b$ с нулевыми температурами на всех сторонах. Математическая постановка: $$ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y), \quad 0<x<a,\;0<y<b, $$ $$ u(0,y)=0,\;u(a,y)=0,\;u(x,0)=0,\;u(x,b)=0. \tag{17} $$ При разностной аппроксимации второй производной по схеме пятиточечного шаблона получаем для узлов (i,j) с шагами $h_x=a/N,\;h_y=b/M$: $$ \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h_x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{h_y^2} = f_{i,j}. \tag{2} $$ При всех граничных условиях $u=0$ на бортах схема (2) замыкается условиями $u_{0,j}=u_{N,j}=u_{i,0}=u_{i,M}=0$.

Для решения вводим разложение решения и правой части в синусоидальные ряды по переменной y (так как $u=0$ при $y=0,b$). Для $j=0,\dots,M$ и $i=0,\dots,N$: $$ u_{i j} = \sum_{m=1}^{M-1} U_{i m}\,\sin\frac{\pi m j}{M}, \quad U_{i m} = \frac{2}{M}\sum_{j=1}^{M-1} u_{i j}\,\sin\frac{\pi m j}{M}, \tag{10-11} $$ $$ f_{i j} = \sum_{m=1}^{M-1} F_{i m}\,\sin\frac{\pi m j}{M}, \quad F_{i m} = \frac{2}{M}\sum_{j=1}^{M-1} f_{i j}\,\sin\frac{\pi m j}{M}. \tag{13-14} $$ Подстановка (10)–(14) в схему (2) даёт систему одномерных уравнений для коэффициентов $U_{i m}$: $$ \frac{U_{i-1,m} - 2U_{i,m} + U_{i+1,m}}{h_x^2} - \frac{\lambda_m}{h_y^2} U_{i,m} = F_{i,m}, \quad \lambda_m = 4\sin^2\frac{\pi m}{2M}, \tag{15} $$ с граничными условиями $U_{0,m}=U_{1,m},\;U_{N,m}=0$ (эквивалентно $u=0$ при $x=a$ и симметричному условию при $x=0$). Каждый разностный краевой случай решается прогонкой, после чего u_{ij} восстанавливается из (10).

Кроме того, общий метод решения можно записать через двойное синусное преобразование Фурье. Сначала по (13)–(14) вычисляются $F_{i m}$. Затем для каждого m,l вводятся формулы $$ U_{i m} = \sum_{l=1}^{N-1} V_{l m}\,\sin\frac{\pi l i}{N}, \quad V_{l m} = \frac{2}{N}\sum_{i=1}^{N-1} U_{i m}\,\sin\frac{\pi l i}{N}, \tag{21-22} $$ $$ F_{i m} = \sum_{l=1}^{N-1} \Phi_{l m}\,\sin\frac{\pi l i}{N}, \quad \Phi_{l m} = \frac{2}{N}\sum_{i=1}^{N-1} F_{i m}\,\sin\frac{\pi l i}{N}. \tag{23-24} $$ После подстановки (21)–(24) в схему (2) получаем для спектральных коэффициентов $V_{l m}$ алгебраические уравнения $$ \Bigl(\frac{\gamma_l}{h_x^2} + \frac{\lambda_m}{h_y^2}\Bigr)V_{l m} = \Phi_{l m}, \quad \gamma_l = 4\sin^2\frac{\pi l}{2N},\;\lambda_m = 4\sin^2\frac{\pi m}{2M}. \tag{25} $$ Решение: $$ V_{l m} = \frac{\Phi_{l m}}{\gamma_l/h_x^2 + \lambda_m/h_y^2}, \tag{26} $$ после чего по (21) и (10) восстанавливаются $U_{i m}$ и $u_{i j}$.

Задача A. Особый случай: на одной стороне прямоугольника задана постоянная (нулевая) температура, остальные три стороны теплоизолированы. Без потери общности считаем $u(a,y)=0$ и $u_x(0,y)=u_y(x,0)=u_y(x,b)=0$. Математически: $$ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y),\quad 0<x<a,\;0<y<b, $$ $$ \frac{\partial u}{\partial x}(0,y)=0,\quad u(a,y)=0,\quad \frac{\partial u}{\partial y}(x,0)=0,\quad \frac{\partial u}{\partial y}(x,b)=0. \tag{1} $$ Схема (2) та же, граничные условия при $x=0,a$ аппроксимируются как $$ u_{0,j} = u_{1,j},\quad u_{N,j}=0,\quad 0\le j\le M, \tag{3} $$ а при $y=0,b$ – как симметричные ($\partial u/\partial y=0$).

В этом случае по y-координате используются разложения по косинусам (Neumann–условия на $y=0,b$): $$ u_{i j} = \frac{U_{i,0}}{2} + \sum_{m=1}^{M-1} U_{i m}\,\cos\frac{\pi m j}{M}, \quad U_{i m} =

\frac{2}{M}\sum_{j=0}^{M} u_{i j}\,\cos\frac{\pi m j}{M}, \tag{4-5} $$ $$ f_{i j} = \frac{F_{i,0}}{2} + \sum_{m=1}^{M-1} F_{i m}\,\cos\frac{\pi m j}{M}, \quad F_{i m} = \frac{2}{M}\sum_{j=0}^{M} f_{i j}\,\cos\frac{\pi m j}{M}. \tag{6-7} $$ Подстановка (4)–(7) в (2) даёт разностную систему: $$ \frac{U_{i-1,m} - 2U_{i,m} + U_{i+1,m}}{h_x^2} + \frac{\rho_m}{h_y^2}\,U_{i,m} = F_{i,m}, \quad \rho_m = 4\sin^2\frac{\pi m}{2M}, \tag{8} $$ с граничными условиями (от (3)) $U_{0,m}=U_{1,m}$, $U_{N,m}=0$ для каждого $m=0,\dots,M$. Система (8) также решается прогонкой. После вычисления коэффициентов $U_{i m}$ решение u_{ij} восстанавливается по (4).

Таким образом, основные формулы для задачи A – уравнение (1), разностная схема (2), условия (3) и формулы косинусного разложения (4)–(8).

Структура Python-классов

В решении задачи **варианта A** можно определить два класса: один отвечает за вычисление и обратное преобразование Фурье, другой – за решение разностной схемы задачи теплопроводности.

- **Класс** `FourierTransform`. Предназначен для вычисления дискретных преобразований Фурье. Параметры: размеры по y (целые M) и по x (N). Методы:
- `forward_cos_y(u)` – вычисляет коэффициенты U_{im} по формуле (5) с помощью дискретного косинус-преобразования (можно использовать `scipy.fftpack.dct` типа I или II), т.е. $$U_{i m} = \frac{2}{M}\sum_{j=0}^M u_{i j}\cos\frac{\pi m j}{M}.$$
- `forward_sin_y(f)` – вычисляет $F_{i m}$ по формуле (7) аналогичным образом (в случае использования синусового ряда по y).

- `inverse_cos_y(U)` – возвращает восстановленное поле u_{ij} по коэффициентам $U_{i m}$ по формуле (4): $$u_{i j} = \frac{U_{i,0}}{2} + \sum_{m=1}^{M-1} U_{i m}\cos\frac{\pi m j}{M}.$$ Аналогично могут быть методы для обратного синусного преобразования (для общего случая). Этот класс инкапсулирует формулы (4)–(7).

- **Класс** `HeatEquationVariantA`. Инициализируется параметрами `a, b, N, M, delta` и функцией источника `f(x,y)`. Свойства:

- `hx = a/N`, `hy = b/M` – шаги сетки.
- `U` – массив для коэффициентов $U_{i m}$.
- `F` – массив для $F_{i m}$.

Методы: 1. `compute_rhs_coeffs()`: вычисляет $F_{i m}$ на сетке по формуле (7) с помощью `FourierTransform.forward_cos_y` (в реализации – например, реализовать суммирование или вызвать `dct`). Это реализует формулы (6)–(7).
2. `solve()`: для каждого $m=0,\dots,M-1$ решает систему (8) (т.е. уравнение трёхдиагональной системы по i) с помощью метода прогонки или `scipy.linalg.solve_banded`. Составляется трёхдиагональная матрица с учетом коэффициентов $1/h_x^2$ на соседях и $(-2/h_x^2 + \rho_m/h_y^2)$ на главной диагонали (формула (8)), и учитываются граничные условия $U_{0,m}=U_{1,m}$, $U_{N,m}=0$. В результате заполняются коэффициенты $U_{i m}$. Этот метод непосредственно использует формулу (8) и граничные условия (3).
3. `reconstruct_solution()`: восстанавливает поле u_{ij} по коэффициентам `U` методом обратного преобразования (4) через `FourierTransform.inverse_cos_y(U)`.

4. `plot_temperature()`: строит 2D-график распределения u_{ij} (например, `matplotlib.pyplot.contourf` или `imshow`). Визуализация опирается на полученную матрицу u после `reconstruct_solution()`.

Таким образом, класс `HeatEquationVariantA` связывает аналитические формулы (1)–(8) с численным решением: инициализация задаёт параметры a,b,N,M,δ, метод `compute_rhs_coeffs` реализует дискретное Фурье-преобразование правой части (формулы (6)–(7)), метод `solve` – решение системы (8)-(3) методом прогонки, а `reconstruct_solution` использует обратное преобразование (4). Метод `plot_temperature` рисует распределение температуры в прямоугольнике.

Оба класса используют библиотеки NumPy/SciPy для операций с массивами и решения СЛАУ, а также matplotlib для построения графика. Связь с теорией прослеживается через указанные формулы: `FourierTransform` – через (4)–(7), `HeatEquationVariantA.solve` – через (8) и (3), а реконструкция решения – через (4).
