Math 320

Rongrong Liu

Final Project First Draft

Finite Difference Method

Finite difference method is also called the method of lines, which as the literal meaning, considers the transient heat transfer and solves the partial differential equations to ordinary differential equations. For convenience, I will use the abbreviation FDM referring to finite difference method. Before rushing into the main part to research on Poisson equation, the first part of the paper will focus on the following problems as warm-up for a deeper exploration:

1. Apply the finite difference method to estimate the solution of the steady state heat equation over a one-dimensional region
2. Apply the finite difference method to estimate the solution of the 2D steady a state heat equation

The most common method to solve heat equation is to use separation of variables.

1. 1D steady state heat equation of a rod of length L

The basic equation: **u(x,t) = X(x) T(t)**

Partial differential Equation: **ut = k uxx**

Boundary Condition: **u(0,t) = u(L,t) = 0**

Initial Condition: **u(x,0) = f(x)**

1. Solution by the method of separation of variables gives that

**= = -λ**

And this generates the solution that

1. **λ = 0**

X(x) = ax + b a,b are real numbers

T(t) = c c is a real number

Therefore, the general solution is **u(x,t) = Ax+B**, A,B are natural numbers.

1. **λ> 0**

**λ=** (**)2**  n is a natural number

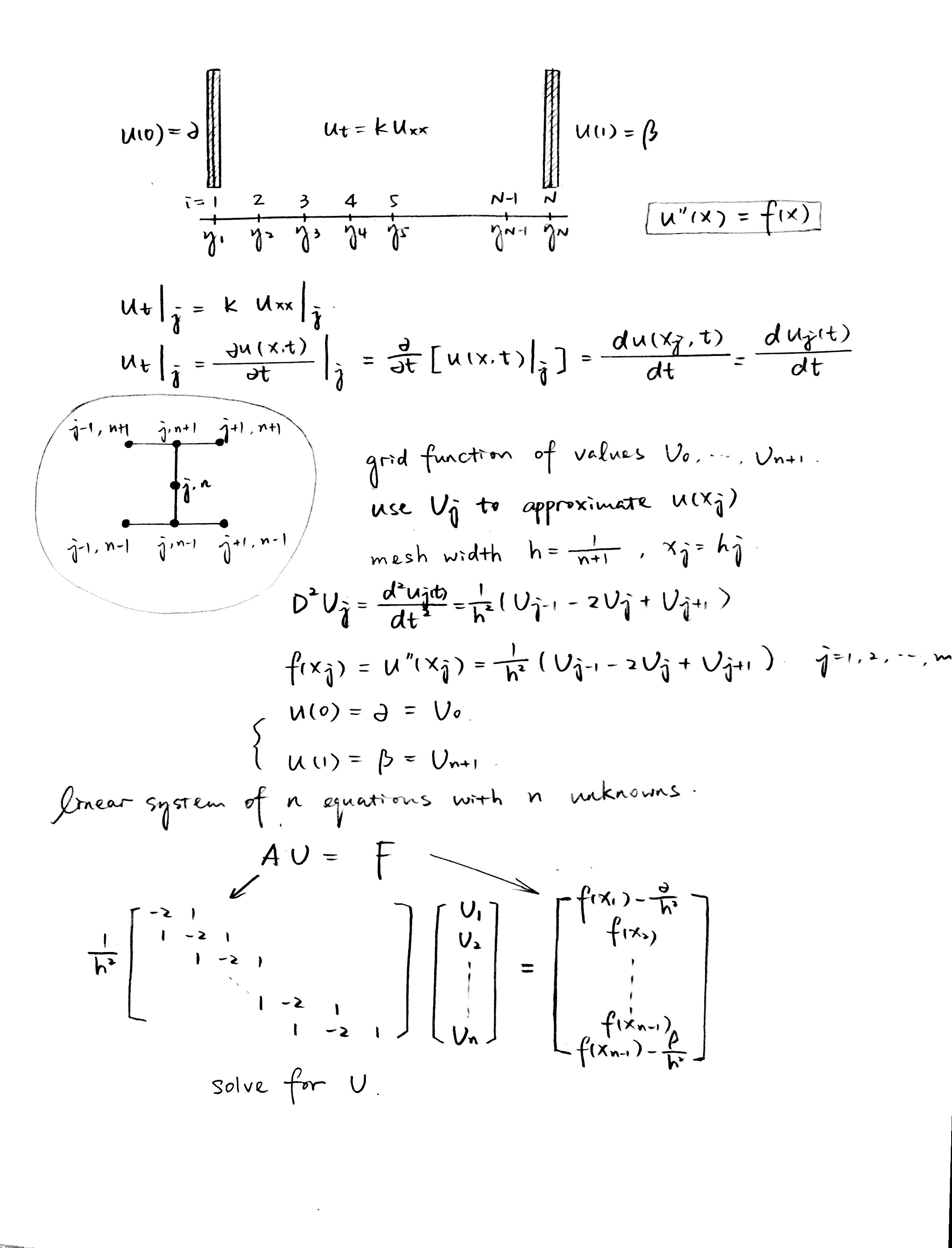
X(x) = c1 cos() + c2 sin() c1 and c2 are real numbers

T(t) =

Therefore, the general solution is u(x,t) = An [sin

1. Solution by the method of FDM

First, “mesh” is a new concept I was exposed to. According the definition on Wikipedia, “mesh” of a partition x0< x1<x2 <….<xn is the length of the longest subinterval, which means max{ (*xi* − *xi*−1) : *i* = 1, ..., *n* }[[1]](#footnote-1).



After we get the matrix [U1 U2 U3 U4……. Un]T, the next step is to explore how well Uj estimates u(xj). The approach is truncation error. (Develop in the next draft)

II. 2D steady state heat equation in a rectangle [0, L]×[0, H]

The basic equation: **u (x, y, t) = X(x) Y(y) T(t)**

Partial differential Equation: **ut = Δu = uxx + uyy = 0**

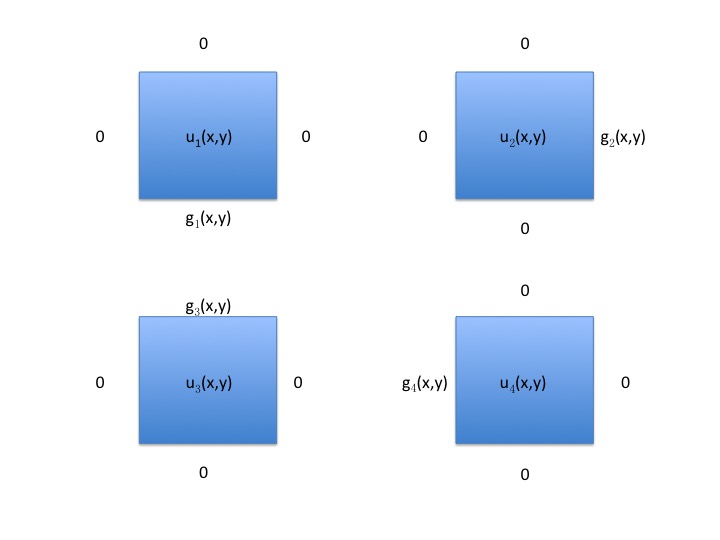
Boundary Condition: **u(0, y) = g1(y)**

**u(L, y) = g2(y)**

**u(x, 0) = f1(x)**

**u(x, H) = f2(x)**

i. Solution by principle of superposition and separation of variables



Basic Idea: u (x, y) = u1(x,y) + u2(x,y) + u3(x,y) + u4(x,y)

Use u4(x,y) as an example: Δu4=0

🡺 **λ=** (**)2**  Y(y) = sin

🡺 X(x) = sinh ()

u4 (x, y) =

g**4**(y) = u4 (0, y) =

An =

The process is similar to calculate u1 (x, y), u2 (x, y) and u3 (x, y). After gaining separate solutions, we use u (x, y) = u1(x,y) + u2(x,y) + u3(x,y) + u4(x,y) to gain the general solution.

1. In the next draft, explore the 2D steady state heat equation using FDM

What’s gonna be on the next draft:

1. Matlab code for these 1D and 2D steady equations
2. Part II of my paper: Possion Equation and its Matlab Code.

1. https://en.wikipedia.org/wiki/Partition\_of\_an\_interval [↑](#footnote-ref-1)