# Bài toán phương trình truyền nhiệt (HeatEquations)

## 1. Giới thiệu bài toán Heat Equations

Một quá trình khuếch tán, ví dụ như khuếch tán nhiệt trong chất rắn hoặc khuếch tán chất tan trong một dung dịch, có thể được mô hình hóa bằng một phương trình đạo hàm riêng.

Xét một hợp chất hóa học có thể hòa tan trong chất lỏng. Hợp chất hóa học này có nồng độ c (concentration) nhất định (số phần tử/m3). Giả sử rằng hợp chất chỉ di chuyển trong dung môi thông qua khuếch tán tự do (để loại đi trường hợp khuếch tán bằng dòng chảy của dung môi). Bây giờ ta đi tìm một phương trình mô tả nồng độ c trên một thể tích nhỏ thông qua định luật Fick (liên quan đến dòng khuếch tán tuyến tính với gradient nồng độ).

Mặt A

J

*Hình 1: Gradient nồng độ ∇c*

Gọi J là thông lượng hay số phần tử đi qua một đơn vị bề mặt trong một thời gian xác định (số phần tử/m2s). Định luật Fick phát biểu rằng thông lượng J và gradient nồng độ c phụ thuộc tuyến tính với nhau theo công thức:

[1]

Với D là hệ số khuếch tán (m2/s).

dx

dz

dy

x

x + dx

Jx(x +dx)

x + dx

*Hình 2: Khối thể tích nhỏ dxdydz.*

Tiếp theo, ta xét một khối thể tích nhỏ dV=dxdydz (hình 2), và tính toán số lượng phần tử khuếch tán vào và ra của khối này. Thực hiện bằng cách tính toán theo ba chiều x, y, z một cách độc lập, sau đó cộng tất cả các kết quả thu được lại để có tổng thông lượng. Số lượng phần tử đi qua khối trong một đơn vị thời gian theo chiều x được cho bởi công thức: (Jx(x)-Jx(x+dx))dydz, tương tự như vậy ta có các công thức cho chiều y và chiều z. Tiếp theo, tổng số phần tử tăng lên trong khối phải bằng số phần tử khuếch tán vào khối. Do đó:

[2]

Trong đó t là thời gian. Chia phương trình [2] cho dV và giả sử rằng dx, dy, dz → 0, ta có:

[3]

Cuối cùng, kết hợp [3] với phương trình Định luật Fick [1] ta được phương trình khuếch tán:

[4]

## 2. Phương pháp số giải bài toán Heat Equations

Ta sẽ giải phương trình Heat Equations [4] bằng phương pháp số, theo đó bao gồm hai bước: rời rạc hóa theo không gian (Spatial discretization) và tích hợp theo thời gian (Time integration).

### *2.1. Phương pháp rời rạc hóa theo không gian*

Xét bài toán Heat Equations trong không gian hai chiều kích thước MxM.

Trong không gian hai chiều, phương trình Heat Equations có dạng:

) [5]

Để giải bài toán trên bằng phương pháp số, trước tiên ta cần chia miền tính toán MxM này thành một lưới điểm. Cho trước kích thước điểm lưới theo chiều x và y tương ứng la dx và dy (dx=dy), ta xác định được số điểm lưới theo chiều x(m) và y(n) như sau :

m = M/dx,

n = M/dy.

Giả sử các điểm lưới được đánh địa chỉ là (i,j), trong đó :

i = 0, 1, 2, …, m-1,

j = 0, 1, 2, …, n-1,

vị trí của điểm lưới (i,j) được xác định như sau :

x = i \* dx,

y = j \* dy.

Từ đây, các điểm liên tục C(x,y) sẽ được đại diện bởi C(i,j).

Cách chia miền tính toán thành một lưới điểm được mô tả như hình 3.

dx

dy

n

m

*Hình 3 chia miền tính toán thành một lưới điểm*

Sau khi đã chia miền tính thành một lưới điểm, ta có thể tính toán vế phải của phương trình (5) sử dụng phương pháp sai phân hữu hạn (Finite Difference Method).

Đặt :

[6]

Để tính đạo hàm bậc 2 của C theo x và y, ta dùng công thức sau :

[7]

Từ (6) và (7) ta có:

[8]

Công thức (8) mô tả cách rời rạc hóa phương trình Heat Equations theo không gian.

### *2.2. Phương pháp tích hợp theo thời gian*

Sau khi rời rạc hóa hệ phương trình Heat Equations theo không gian ta sẽ thu được một hệ phương trình vi phân thường (Ordinary Differential Equations) phụ thuộc vào thời gian t có dạng :

[9]

Có nhiều phương pháp để giải phương trình vi phân thường (3.8), trong luận văn này chúng tôi sẽ sử dụng phương pháp Euler thuận (Forward Euler).

Phương pháp Euler thuận giải phương trình vi phân thường có công thức như sau:

[10]

Trong đó n là bước tính toán thứ n, ht là độ dài bước thời gian.

Công thức Euler thuận có thể được tính toán lặp như sau:

……. [11]

Trong đó,C­0 là giá trị khởi tạo ban đầu.

**3. Yêu cầu**

* **Cài đặt chương trình Matlab hoặc C để giải bài toán Heat Equation.**
* **Dùng matlab để vẽ hình, tạo clip mô phỏng kết quả đạt được**

**Các tham số đầu vào sinh viên tự chọn một cách hợp lý.**

VD các tham số đầu vào :

#define m 20

#define n 20

#define T 1

#define dt 0.01

#define dx 0.1

#define dy 0.1

#define D 0.1

**Giá trị ban đầu sinh viên tự chọn.**

Ví dụ hàm khởi tạo giá trị ban đầu:

void KhoiTao(float \*C)

{

int i,j;

for ( i = 0 ; i < m ; i++ )

for ( j = 0 ; j < n ; j++ ){

if (i==0)

\*(C+i\*n+j) = 1.0;

else

\*(C+i\*n+j) = 0.0;

}

}

**Điều kiện biên :** Trong phương trình (8), khi tính FD tại các điểm lưới i=0, i=m-1, j=0, j=n-1 thì cần giá trị của C tại các điểm nằm ngoài miền dữ liệu sẵn có. Để có đủ thông tin, chúng ta sử dụng điều kiện biên. Giá trị của điều kiện biên phụ thuộc vào từng bài toán. Ví dụ bạn có thể chọn điều kiện biên như sau :

- Cố định giá trị: C(0,j) = 1, C(m-1,j) = 0 với mọi j

- Tuần hoàn : C(i,-1) = C(i,n-1) ; C(i,n) = C(i,0) với mọi i