Phần 2: mô phỏng máy tính

Modeling, simulation and optimization for chemical process

Instructor: Hoang Ngoc Ha

Email: ha.hoang@hcmut.edu.vn

Bộ môn QT&TB

Introduction

Numerical Analysis

Computer Programming

SIMULATION

Computer simulation

- Some simulation techniques for solving some of the systems of equations
  - Solution of (nonlinear) algebraic equations
  - Ordinary differential equations (ODEs)
  - Partial differential equations (PDEs)
- Numerical methods
  - Iterative methods
  - Discrete difference methods
  - Femlab, Fortran, Ansys… **Matlab/Simulink**

Computer simulation

- Computer programming
  - Assume that you know some computer programming language
  - We are not interested in generating the most efficient and elegant code but in solving problems (from point of view of engineers)
    - Including extensive comment statements
    - Use of symbols (the same ones in the equations describing the systems)
    - Debugging (for mistakes in coding and/or in logic)
    - …

Computer simulation

Example: We are given the pressure ***P*** and the liquid composition x. We want to find the bubblepoint temperature and the vapor composition as discussed in Sec. 2.2.6. For simplicity let us assume a binary system of components 1 and 2. Component 1 is the more volatile, and the mole fraction of component 1 in the liquid is x and in the vapor is y. Let us assume also that the system is ideal: Raoult's and Dalton's laws apply.

Computer simulation

The partial pressures of the two components ($\mathcal{P}_1$ and $\mathcal{P}_2$) in the liquid and vapor phases are :

In liquid: $$\mathcal{P}_1 = xP_1^s \qquad \mathcal{P}_2 = (1 - x)P_2^s \tag{4.1}$$

In vapor: $$\mathcal{P}_1 = yP \qquad \mathcal{P}_2 = (1 - y)P \tag{4.2}$$

where P_j^s = vapor pressure of pure component j which is a function of only temperature

$$\ln P_1^s = \frac{A_1}{T} + B_1 \qquad \ln P_2^s = \frac{A_2}{T} + B_2 \tag{4.3}$$

Equating partial pressures in liquid and vapor phases gives

$$P = xP_1^s + (1 - x)P_2^s \tag{4.4}$$

$$y = \frac{xP_1^s}{P} \tag{4.5}$$

Our convergence problem is to find the value of temperature T that will satisfy Eq. (4.4).

Computer simulation

The procedure is as follows:

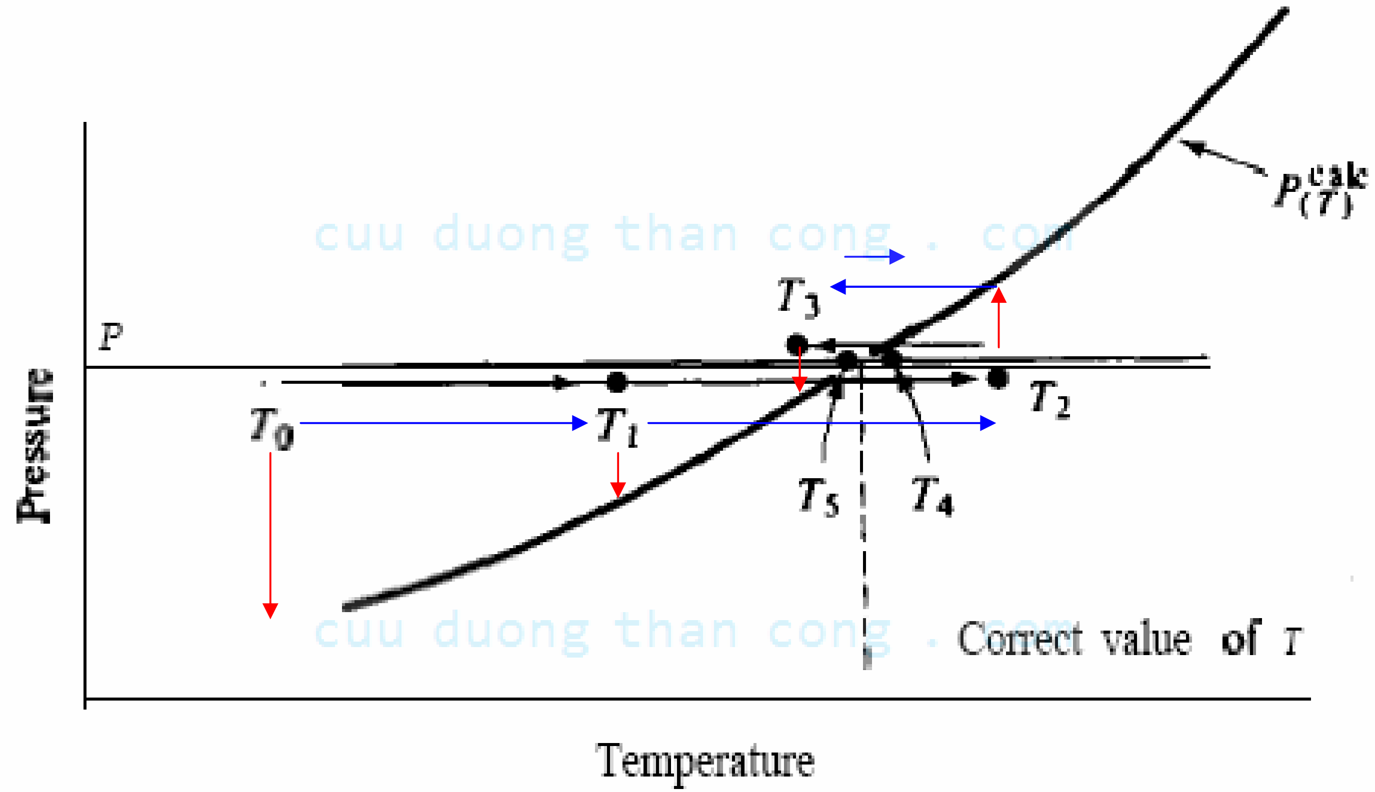
1. Guess a temperature T.
2. Calculate the vapor pressures of components 1 and 2 from Eq. (4.3).
3. Calculate a total pressure P^{calc} using Eq. (4.4).

$$P^{calc} = xP^{s}_{1(T)} + (1 - x)P^{s}_{2(T)} \tag{4.6}$$

4. Compare P^{calc} with the actual total pressure given, P. If it is sufficiently close to P (perhaps using a relative convergence criterion of 10^{-6}), the guess T is correct. The vapor composition can then be calculated from Eq. (4.5).
5. If P^{calc} is greater than P, the guessed temperature was too high and we must make another guess of T that is lower. If P^{calc} is too low, we must guess a higher T.

Computer simulation

- Interval halving (chia đôi khoảng)



Computer simulation

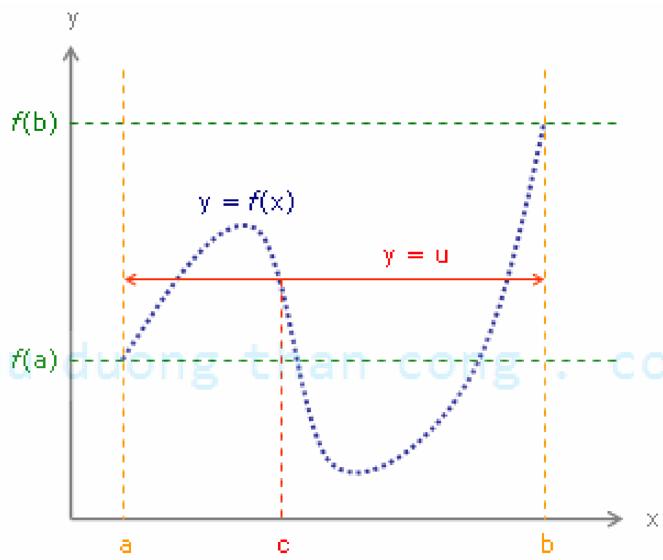
- This problem can be formulated under the following form:

$$f(x) = 0, x \in R$$

- The goal is to find the solution of this nonlinear equations (**in ONE VARIABLE**)
- Tools (Iterative methods)
  - Bisection method (phương pháp phân đoạn)
  - Newton's (or Newton-Raphson) method

Iterative method

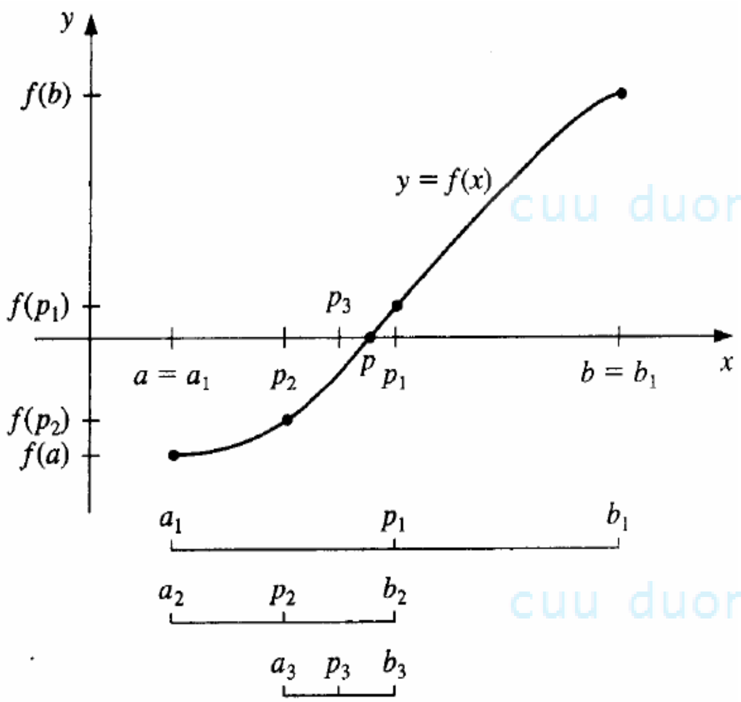
- **Intermediate value theorem**
  - If *f* is a real-valued continuous function on the interval [*a*, *b*], and *u* is a number between *f*(*a*) and *f*(*b*), then there is a $c \in [a, b]$ such that *f*(*c*) = *u*



If *f(a)* and *f(b)* are of opposite sign, there exist a number *p* in [*a, b*] with *f(p)=0*

Iterative method

- Bisection method



To find a solution to $f(x) = 0$ given the continuous function f on the interval $[a, b]$, where $f(a)$ and $f(b)$ have opposite signs:

INPUT endpoints a, b; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$;
$FA = f(a)$.

Step 2 While $i \le N_0$ do Steps 3–6.

Step 3 Set $p = a + (b - a)/2$; *(Compute p_i.)*
$FP = f(p)$.

Step 4 If $FP = 0$ or $(b - a)/2 < TOL$ then
OUTPUT (p); *(Procedure completed successfully.)*
STOP.

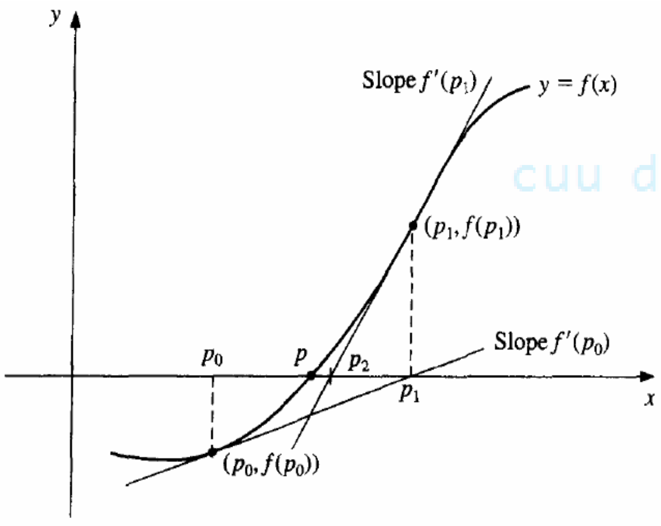
Step 5 Set $i = i + 1$.

Step 6 If $FA \cdot FP > 0$ then set $a = p$; *(Compute a_i, b_i.)*
$FA = FP$
else set $b = p$.

Step 7 OUTPUT ('Method failed after N_0 iterations, N_0 =', N_0);
(The procedure was unsuccessful.)
STOP.

Iterative method

- Newton's method



To find a solution to $f(x) = 0$ given an initial approximation p_0:

INPUT initial approximation p_0; tolerance TOL; maximum number of iterations N_0.

OUTPUT approximate solution p or message of failure.

Step 1 Set $i = 1$.

Step 2 While $i \le N_0$ do Steps 3–6.

Step 3 Set $p = p_0 - f(p_0)/f'(p_0)$. *(Compute p_i.)*

Step 4 If $|p - p_0| < TOL$ then
OUTPUT (p); *(The procedure was successful.)*
STOP.

Step 5 Set $i = i + 1$.

Step 6 Set $p_0 = p$. *(Update p_0.)*

Step 7 OUTPUT ('The method failed after N_0 iterations, N_0 =', N_0);
(The procedure was unsuccessful.)
STOP.

Numerical solutions of nonlinear systems of equations (**of SEVERAL VARIABLES**)➔ (See Ref.)

Computer simulation

- Interpolation and polynomial approximation
  - **Interpolation and the Lagrange polynomial**
  - Cubic spline interpolation
  - …
- Numerical differentiation and intergration
  - Numerical differentiation
  - Richardson's extrapolation
  - …

(See Ref.)

Numerical intergration of Ordinary Differential Equations (ODEs)

The first part of this chapter is concerned with approximating the solution $y(t)$ to a problem of the form

$$\frac{dy}{dt} = f(t, y), \quad \text{for } a \leq t \leq b,$$

subject to an initial condition

$$y(a) = \alpha.$$

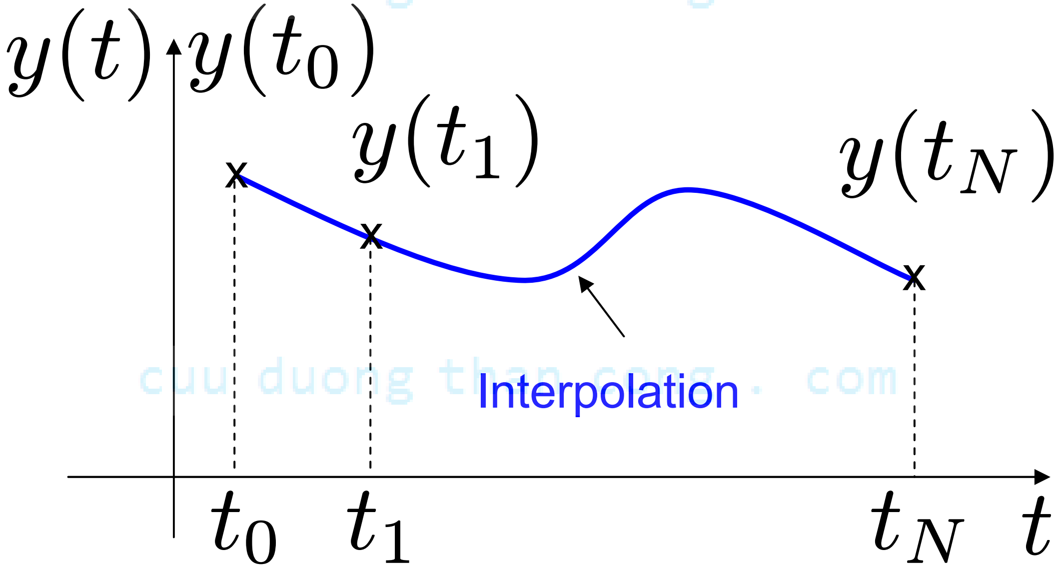
In actuality, a continuous approximation to the solution $y(t)$ will not be obtained; instead, approximations to y will be generated at various values, called **mesh points**, in the interval $[a, b]$. Once the approximate solution is obtained at the points, the approximate solution at other points in the interval are found by interpolation.

Numerical intergration of Ordinary Differential Equations (ODEs)

We first make the stipulation that the mesh points are equally distributed throughout the interval $[a, b]$. This condition is ensured by choosing a positive integer N and selecting the mesh points

$$t_i = a + ih, \quad \text{for each } i = 0, 1, 2, \ldots, N.$$

The common distance between the points $h = (b - a)/N$ is called the **step size**.



Numerical intergration of Ordinary Differential Equations (ODEs)

- Tools:
  - Euler's method

  - Higher-Order Taylor methods
  - Runge-Kutta methods

  - …

Numerical intergration of Ordinary Differential Equations (ODEs)

- Euler's method

The object of the method is to obtain an approximation to the well-posed initial-value problem

$$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha. \tag{5.6}$$

Numerical intergration of Ordinary Differential Equations (ODEs)

- Euler's method

We will use Taylor's Theorem to derive Euler's method. Suppose that $y(t)$, the unique solution to (5.6), has two continuous derivatives on $[a, b]$, so that for each $i = 0, 1, 2, \ldots, N-1$,

$$y(t_{i+1}) = y(t_i) + (t_{i+1} - t_i)y'(t_i) + \frac{(t_{i+1} - t_i)^2}{2} y''(\xi_i),$$

for some number ξ_i in (t_i, t_{i+1}). Since $h = t_{i+1} - t_i$, we have

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(\xi_i),$$

and, since $y(t)$ satisfies the differential equation (5.6),

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2} y''(\xi_i). \qquad (5.7)$$

Euler's method constructs $w_i \approx y(t_i)$, for each $i = 1, 2, \ldots, N$, by deleting the remainder term. Thus, Euler's method is

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hf(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1. \qquad (5.8)$$

Equation (5.8) is called the **difference equation** associated with Euler's method.

Numerical intergration of Ordinary Differential Equations (ODEs)

- Example

$$y' = y - t^2 + 1, t \in [0 \quad 2]$$
$$y(0) = 0.5$$

P/p Euler *n=10?*

Approximate solution?

$$n = 10 \Rightarrow h = \frac{b-a}{n} = 0.2$$

Exact solution?

$$y(t) = -0.5\exp(t) + (t+1)^2$$

Numerical intergration of Ordinary Differential Equations (ODEs)

- Local truncation error

Definition

The difference method

$$w_0 = \alpha$$

$$w_{i+1} = w_i + h\phi(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1,$$

has **local truncation error**

$$\tau_{i+1}(h) = \frac{y_{i+1} - (y_i + h\phi(t_i, y_i))}{h} = \frac{y_{i+1} - y_i}{h} - \phi(t_i, y_i),$$

for each $i = 0, 1, \ldots, N-1$.

The local truncation error in Euler's method is $O(h)$

Numerical intergration of Ordinary Differential Equations (ODEs)

- Higher-Order Taylor methods

Suppose the solution $y(t)$ to the initial-value problem

$$y' = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha,$$

has $(n+1)$ continuous derivatives. If we expand the solution, $y(t)$, in terms of its nth Taylor polynomial about t_i and evaluate at t_{i+1}, we obtain

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \cdots + \frac{h^n}{n!}y^{(n)}(t_i) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi_i), \quad (5.15)$$

for some ξ_i in (t_i, t_{i+1}).

Successive differentiation of the solution, $y(t)$, gives

$$y'(t) = f(t, y(t)),$$

$$y''(t) = f'(t, y(t)),$$

and, in general,

$$y^{(k)}(t) = f^{(k-1)}(t, y(t)).$$

Substituting these results into Eq. (5.15) gives

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2}f'(t_i, y(t_i)) + \cdots \quad (5.16)$$

Numerical intergration of Ordinary Differential Equations (ODEs)

- Higher-Order Taylor methods

Substituting these results into Eq. (5.15) gives

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \frac{h^2}{2}f'(t_i, y(t_i)) + \cdots + \frac{h^n}{n!}f^{(n-1)}(t_i, y(t_i)) + \frac{h^{n+1}}{(n+1)!}f^{(n)}(\xi_i, y(\xi_i)). \quad (5.16)$$

The difference-equation method corresponding to Eq. (5.16) is obtained by deleting the remainder term involving ξ_i. This method is called the

Taylor method of order n:

$$w_0 = \alpha,$$
$$w_{i+1} = w_i + hT^{(n)}(t_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1, \quad (5.17)$$

where

$$T^{(n)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2}f'(t_i, w_i) + \cdots + \frac{h^{n-1}}{n!}f^{(n-1)}(t_i, w_i).$$

Note that Euler's method is Taylor's method of order one.

Numerical intergration of Ordinary Differential Equations (ODEs)

- Runge-Kutta methods

The Taylor methods outlined in the previous section have the desirable property of high-order local truncation error, but the disadvantage of requiring the computation and evaluation of the derivatives of $f(t, y)$. This is a complicated and time-consuming procedure for most problems, so the Taylor methods are seldom used in practice.

Runge-Kutta methods have the high-order local truncation error of the Taylor methods while eliminating the need to compute and evaluate the derivatives of $f(t, y)$.

Numerical intergration of Ordinary Differential Equations (ODEs)

- Runge-Kutta methods
  - Xây dựng công thức tính w_{i+1} theo w_i mà không phải đạo hàm « tay », cần xấp xỉ $T^{(k)}$ mà không dùng đạo hàm với $O(h^k)$
    - Minh họa qua *k*=2

$$T^{(2)}(t,y) = f(t,y) + \frac{h}{2} f'(t,y)$$

$$f'(t,y) = f'_t(t,y) + f'_y(t,y)y'(t)$$

$$f(t,y)$$

Numerical intergration of Ordinary Differential Equations (ODEs)

Như vậy

$$T^{(2)}(t,y) = f(t,y) + \frac{h}{2} f_t'(t,y) + \frac{h}{2} f_y'(t,y) f(t,y)$$

Cần tìm a_1, α_1, β_1 với sai số $O(h^2)$ để

$$a_1 f(t+\alpha_1, y+\beta_1) \simeq T^{(2)}(t,y)$$

$$f(t+\alpha_1, y+\beta_1) \simeq f(t,y) + f_t'(t,y)\alpha_1 + f_y'(t,y)\beta_1$$

Numerical intergration of Ordinary Differential Equations (ODEs)

Cần chọn

$$a_1 f(t,y) + a_1\alpha_1 f'_t(t,y) + a_1\beta_1 f'_y(t,y) = f(t,y) + \frac{h}{2} f'_t(t,y) + \frac{h}{2} f'_y(t,y) f(t,y)$$

Đồng nhất hai vế

$$\begin{cases} a_1 = 1 \\ a_1\alpha_1 = \frac{h}{2} \\ a_1\beta_1 = \frac{h}{2} f(t,y) \end{cases} \qquad \begin{cases} a_1 = 1 \\ \alpha_1 = \frac{h}{2} \\ \beta_1 = \frac{h}{2} f(t,y) \end{cases}$$

Numerical intergration of Ordinary Differential Equations (ODEs)

$$w_{i+1} = w_i + hT^{(2)}(t_i, w_i)$$

$$w_{i+1} = w_i + h\left[f\left(t_i + \frac{h}{2}, w_i + \frac{h}{2} f(t_i, w_i)\right)\right]$$

Sơ đồ trung điểm (R_K bậc 2)

$$\begin{cases} w_0 = \alpha \\ k_1 = \frac{h}{2} f(t_i, w_i) \\ k_2 = hf(t_i + \frac{h}{2}, w_i + k_1) \\ w_{i+1} = w_i + k_2 \end{cases}$$

Numerical intergration of Ordinary Differential Equations (ODEs)

Sơ đồ R_K bậc 4

$$\begin{cases} w_0 = \alpha \\ k_1 = hf(t_i, w_i) \\ k_2 = hf(t_i + \frac{h}{2}, w_i + \frac{k_1}{2}) \\ k_3 = hf(t_i + \frac{h}{2}, w_i + \frac{k_2}{2}) \\ k_4 = hf(t_i + h, w_i + k_3) \\ w_{i+1} = w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases}$$

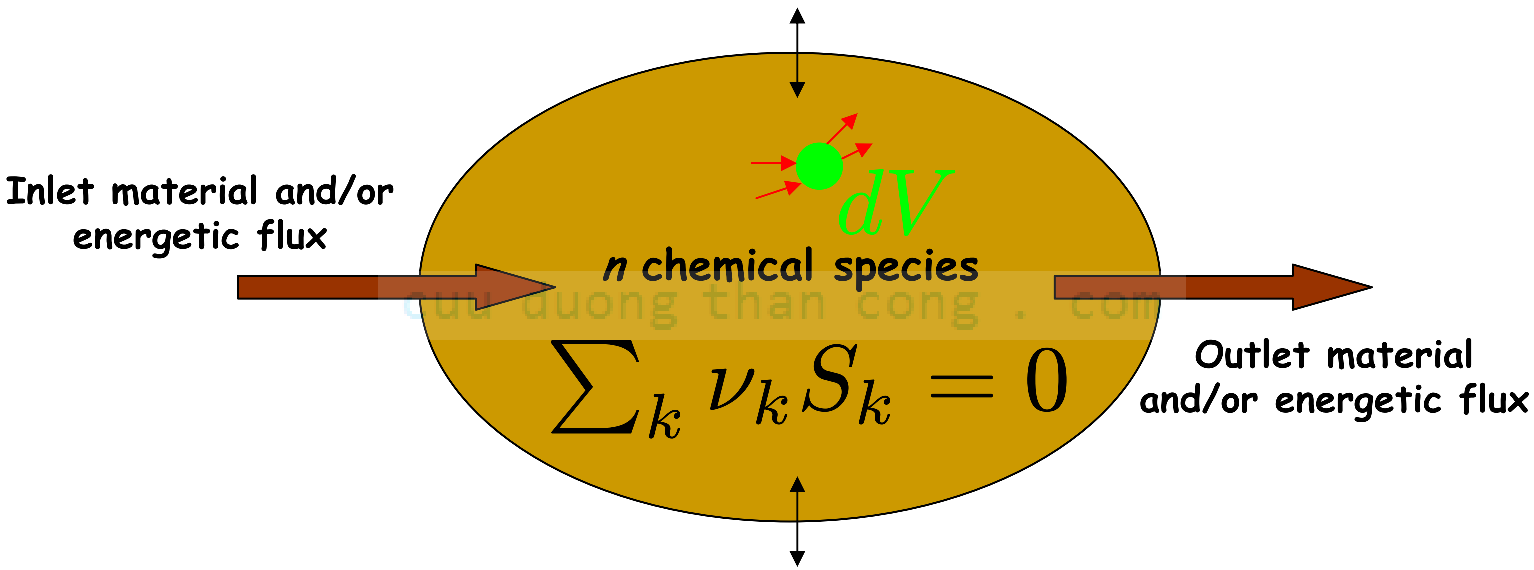
cuu duong than cong . com

Về nhà tự đọc R_K cho hệ và viết chương trình R_K cho hệ

Numerical intergration of Partial Differential Equations (PDEs)

- Click here

Giới thiệu chung



$u = u(x \ldots, t)$ **một đại lượng vật lý của hệ khảo sát**

Profile

$\{C, T \ldots\}$ **HỆ PHÂN BỐ**

Local observation

Ba dạng phương trình đạo hàm riêng cơ bản

Động học biến hệ thống $u = u(x \ldots, t)$ có thể thuộc về các dạng phương trình sau:

- **Phương trình elliptic (tĩnh-static)**

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$$

- **P/t parabolic (b/toán truyền nhiệt)**

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}$$

- Phương trình hyperbolic (b/toán truyền sóng)

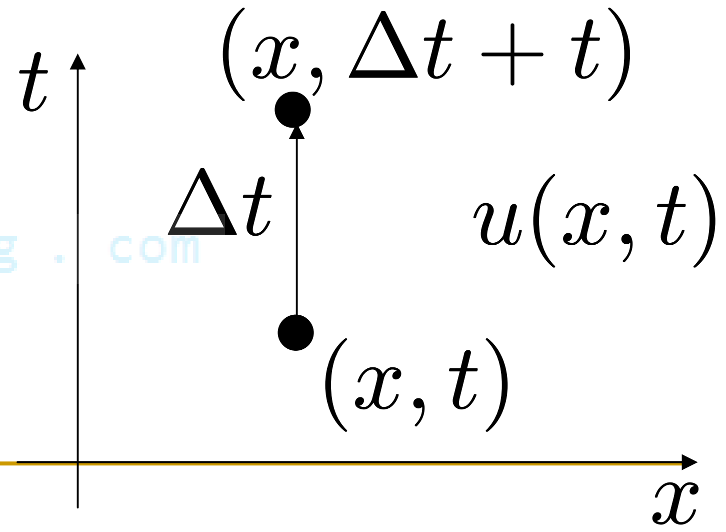
$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}$$

Ba dạng phương trình đạo hàm riêng cơ bản

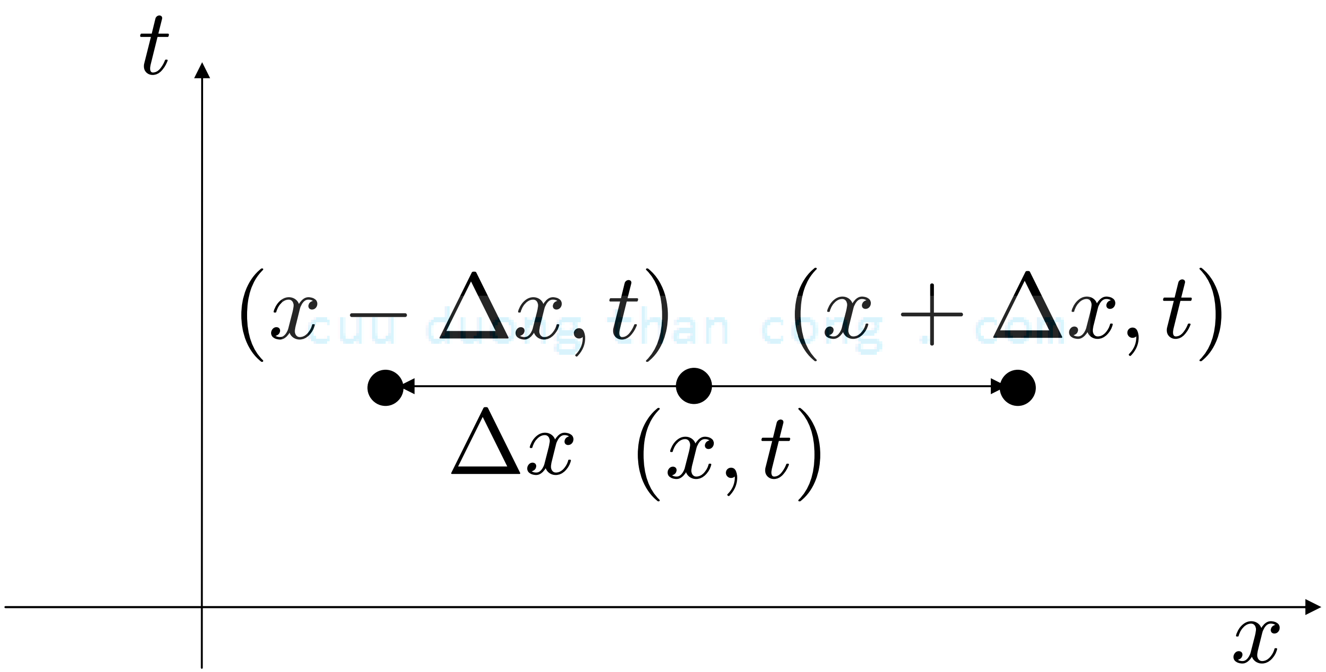
- Phương pháp tìm nghiệm
  - Phương pháp giải tích
  - Phương pháp số

Ý tưởng: xấp xỉ sai phân các đạo hàm riêng tại các điểm rời rạc (kg,tg) và tính giá trị của $u = u(x \ldots, t)$ **tại đó**

$$\frac{\partial u(x,t)}{\partial t} \approx \frac{u(x,t+\Delta t) - u(x,t)}{\Delta t}$$

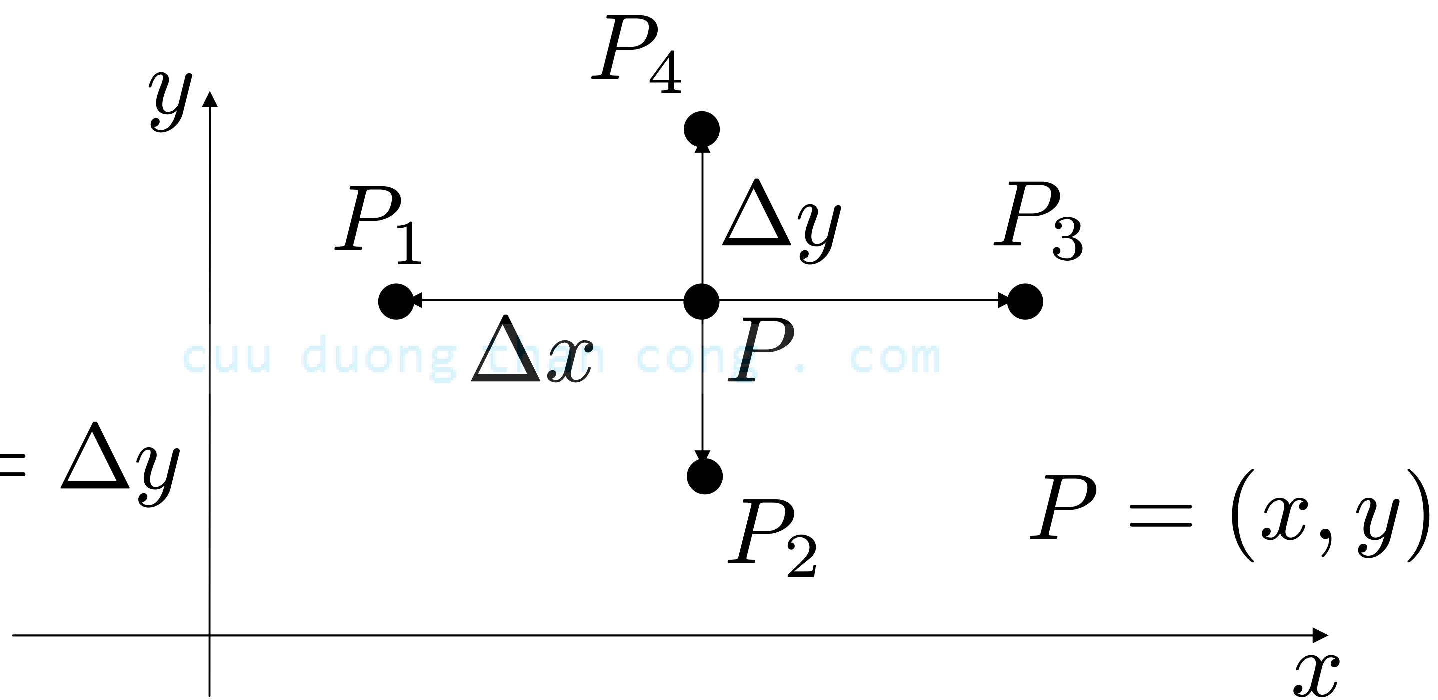


Xấp xỉ sai phân



$$\frac{\partial^2 u(x,t)}{\partial x^2} \approx \frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{(\Delta x)^2}$$

Xấp xỉ sai phân



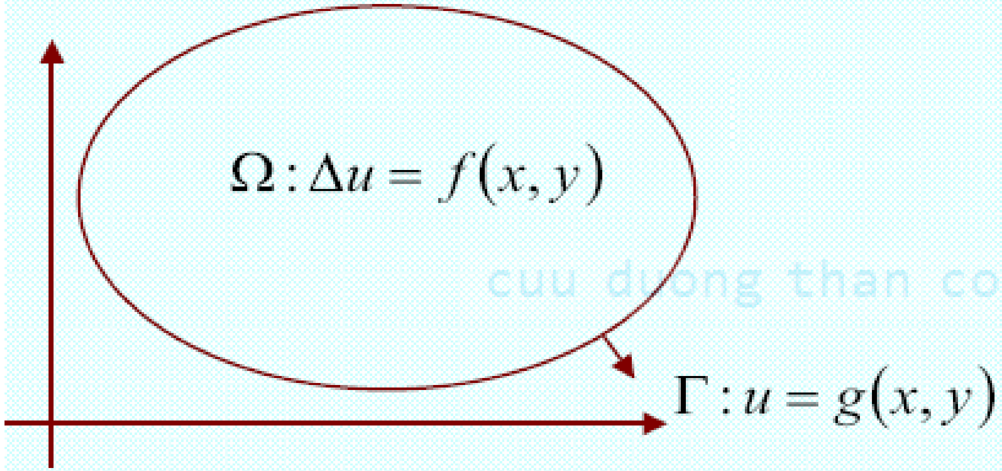
$$h = \Delta x = \Delta y$$

$$\frac{\partial^2 u(P)}{\partial x^2} + \frac{\partial^2 u(P)}{\partial y^2} \approx \frac{u(P_1)+u(P_2)+u(P_3)+u(P_4)-4u(P)}{h^2}$$

BÀI TOÁN ELLIPTIC

- Bài toán elliptic với điều kiện biên Dirichlet

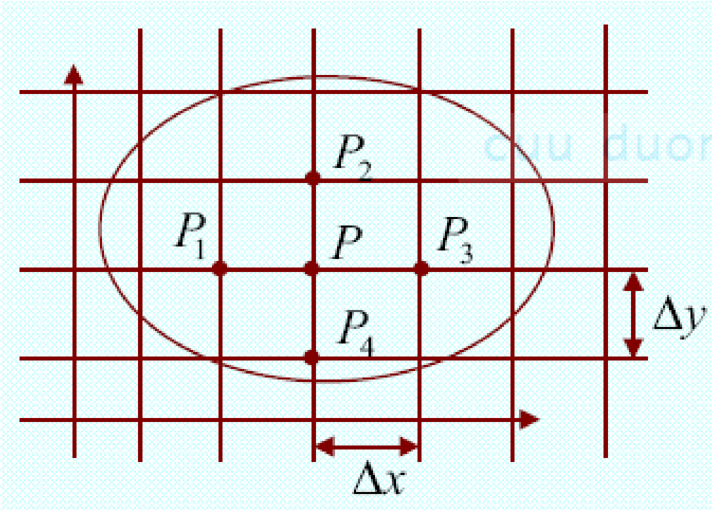
$$\begin{cases} \Delta u = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = f(x,y), (x,y) \in \Omega \subset R^2 \\ u(x,y) = g(x,y), (x,y) \in \Gamma \end{cases}$$



Giải bằng sai phân hữu hạn: Chia nhỏ Ω. Tính xấp xỉ giá trị nghiệm u tại các điểm chia

BÀI TOÁN ELLIPTIC

- Phân hoạch và tạo lưới Ω: chia nhỏ Ω bởi các đường thẳng // với *Ox* và *Oy*



Tạo lưới bước chia cách đều

$$h = \Delta x = \Delta y$$

Kí hiệu ***P_1, P_2, P_3*** *và* ***P_4*** *là 4 điểm rời rạc x/q* ***P***

$$\Delta u(P) \approx \frac{u(P_1)+u(P_2)+u(P_3)+u(P_4)-4u(P)}{h^2}$$

Lần lượt thay $P_k = (x_k, y_k)$ vào phương trình elliptic, sử dụng công thức xấp xỉ & điều kiên biên ➡ **Hệ PTTT ẩn** $u_k = u(P_k)$

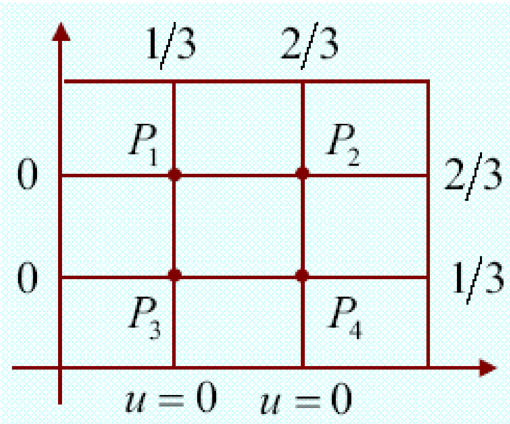
BÀI TOÁN ELLIPTIC

- Example

Giải bài toán
$$\begin{cases} \partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = 0, \ 0 < x < 1, \ 0 < y < 1 \\ u(x,0) = 0, \ u(x,1) = x \quad 0 \le x \le 1 \\ u(0,y) = 0, \ u(1,y) = y, \ 0 \le y \le 1 \end{cases}$$

bởi lưới bước chia cách đều h = 1/3 trên 2 trục Ox và Oy

Lưới 4 nút ẩn ⇨ đánh số 4 giá trị cần tìm. SD PTSP & Giá trị trên biên ⇨ Hệ PTTT



Nút 1: $1/3 + u_3 + u_2 - 4u_1 = 0$

Nút 2: $2/3 + u_4 + u_1 + 2/3 - 4u_2 = 0$

Nút 3: $u_1 + u_4 - 4u_3 = 0$

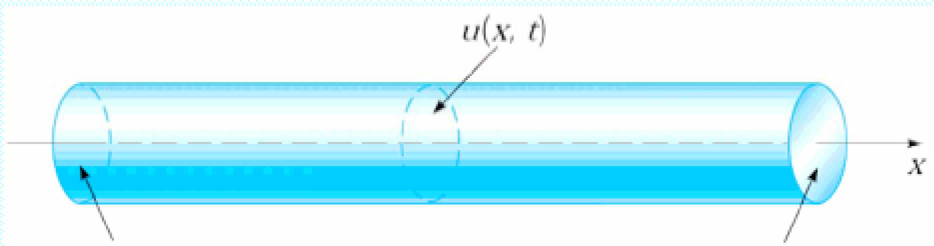
Nút 4: $u_2 + u_3 + 1/3 - 4u_4 = 0$

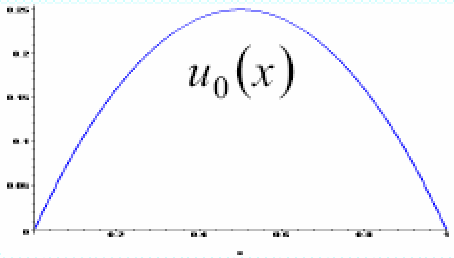
Mô phỏng Matlab

BÀI TOÁN PARABOLIC (mô hình truyền nhiệt)

Bài toán truyền nhiệt: Xác định nhiệt độ theo thời gian tại điểm bất kỳ trên thanh kim loại độ dài 1, có nhiệt độ ban đầu ở thời điểm t = 0: $u_0(x)$ và 2 đầu thanh luôn có nhiệt độ = 0

Thanh kim loại $0 \le x \le 1$: **Ban đầu (t = 0):**

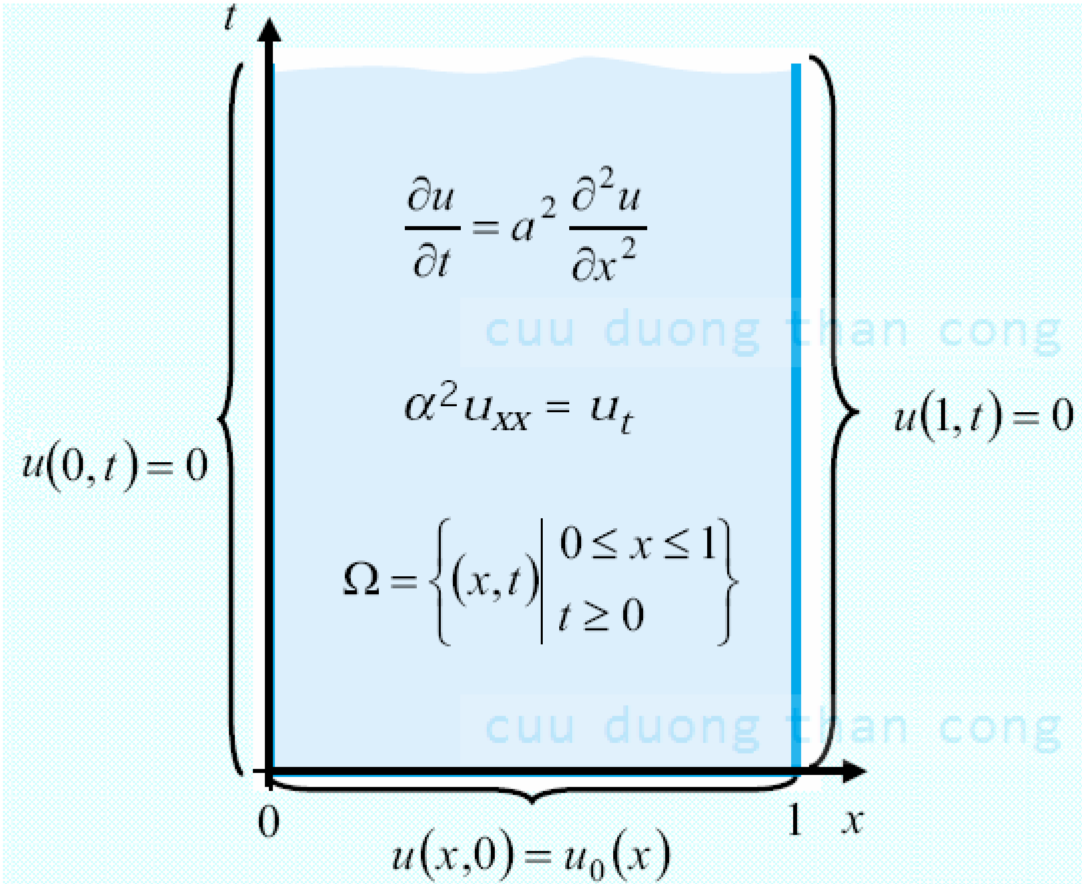




$x = 0$ $x = 1$ $x = 0$ $x = 1$

$$\Rightarrow \frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}, 0 < x < 1, t > 0 \qquad u(0,t) = u(1,t) = 0 \qquad u(x,0) = u_0(x)$$

BÀI TOÁN PARABOLIC (mô hình truyền nhiệt)

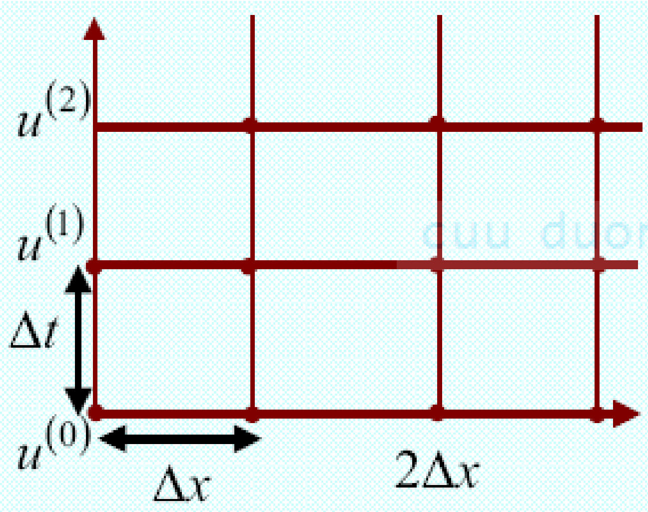


ĐK biên + ĐK đầu → Giá trị u trên biên: ĐK Dirichlet

Phương trình đạo hàm riêng + ĐK Dirichlet (giá trị biên) → ∃ nghiệm & duy nhất.

BÀI TOÁN PARABOLIC (mô hình truyền nhiệt)

- Phân hoạch và xấp xỉ



Chia cách đều bước Δx và Δt

Mức thứ *k*: $t = k\Delta t$

$$u^{(k)} = (\ldots u_i^k \ldots)$$

Mức thứ *0*: $u^{(0)}$ **đã biết**

Mức thứ *1*: $u^{(1)}$ **chưa biết**

Từ giá trị $u^{(k)}$ ở mức thời gian thứ k đã biết tính giá trị $u^{(k+1)}$?

Nguyên tắc: Thay mốc (x_i, t_j) vào PT & xấp xỉ các ĐH riêng

Xấp xỉ đạo hàm riêng cấp 1: $\dfrac{\partial u(x,t)}{\partial t}$ **TIẾN - LÙI**

BÀI TOÁN PARABOLIC (mô hình truyền nhiệt)

- Ví dụ

$$\begin{cases} \dfrac{\partial u}{\partial t}(x,t) - \dfrac{\partial^2 u}{\partial x^2}(x,t) = 0, \ \ 0 < x < 1, t > 0 \\ u(0,t) = u(1,t) = 0, t > 0; \ \ u(x,0) = \sin \pi x \end{cases} \Rightarrow u(x,t) = e^{-\pi^2 t} \sin \pi x$$

$u(x,0.5)$, $\Delta x = 0.1$: a/ Hiện, $\Delta t = 0.01$ & 0.0005 b/ Ẩn, $\Delta t = 0.01$

Δx	Δt	$\lvert u(x, 0.5) - u^{(k)} \rvert$
0.1	$0.01 \Rightarrow k = 50$	$8.2 \times 10^{7} \rightarrow 2.63 \times 10^{8}$
	$0.0005 \Rightarrow k = 1000$	$6.4 \times 10^{-5} \rightarrow 2.1 \times 10^{-4}$

Sơ đồ hiện (s/p tiến)

Δx	Δt	$\lvert u(x, 0.5) - u^{(k)} \rvert$
0.1	$0.01 \Rightarrow k = 50$	$6.8 \times 10^{-4} \rightarrow 2.2 \times 10^{-3}$

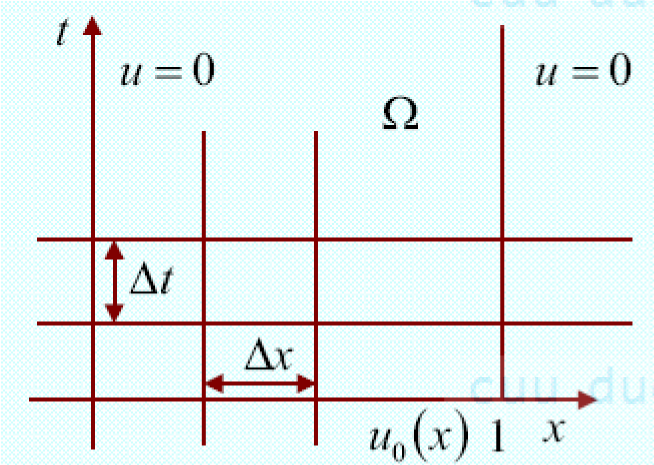
Sơ đồ ẩn (s/p lùi)

BÀI TOÁN PARABOLIC (mô hình truyền nhiệt tổng quát)

$$\frac{\partial u}{\partial t}(x,t) - a^2\frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t),\ 0 < x < 1,\ t > 0$$

$$u(0,t) = u(1,t) = 0, t > 0 \qquad u(x,0) = u_0(x),\ 0 \le x \le 1$$

Miền $\Omega = \{(x,t) | 0 \le x \le 1, t \ge 0\}$



Phân hoạch Ω : Lưới theo x độ dài Δx, theo t độ dài Δt $\Rightarrow$ Các đường thẳng x = i Δx, t = k Δt

Xấp xỉ $\frac{\partial u}{\partial t}, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}$ **& ĐK biên, đầu** ➡ **Giá trị *u* tại các điểm lưới**

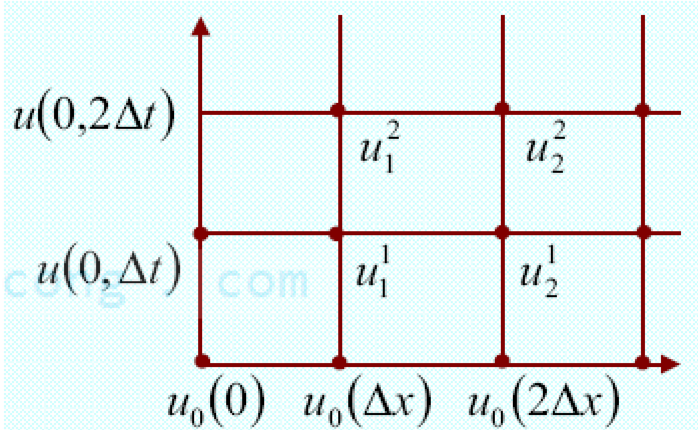
Sai phân tiến & Sai phân lùi

BÀI TOÁN PARABOLIC (mô hình truyền nhiệt)

Ox: Các đoạn độ dài Δx = l/(n+1)

Ot (t > 0) : Các đoạn độ dài Δt

Nút (iΔx, kΔt) ⇒ u(iΔx, kΔt)= u_i^k

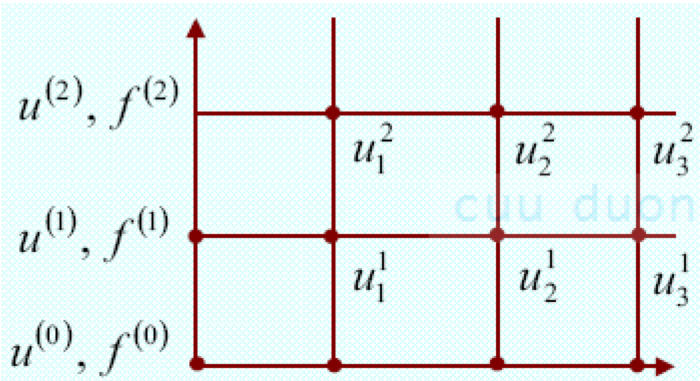


Điều kiện biên: THUẦN NHẤT (u = 0) tại x = 0, x = l và điều kiện ban đầu (t = 0):

$$\underbrace{u_0^k = u_{n+1}^k = 0\,,\, k \geq 0}_{\text{Điều Kiện Biên } (x=0\ \&\ x=\ell)} \;;\; \underbrace{u_i^0 = u_0(i\Delta x) = u_{0i}\,,\, i = 0 \to n+1}_{\text{Điều Kiện Ban Đầu } (t=0)}$$

BÀI TOÁN PARABOLIC (mô hình truyền nhiệt)

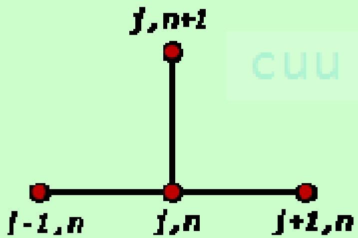
Kí hiệu $u^{(k)} = [u_1^k \dots u_n^k], f^{(k)} = [f_1^k \dots f_n^k]$



$$\frac{\partial u}{\partial t} - a^2 \frac{\partial^2 u}{\partial^2 x} = f(x,t)$$

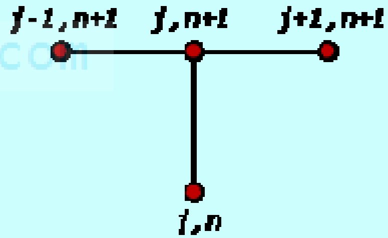
Biết $u^{(0)}$, $f^{(n)}$ $\forall$ n ≥ 0. Giả sử biết $u^{(n)}$ $\Rightarrow$ Cần tính $u^{(n+1)}$

Sai phân tiến (sơ đồ hiện)



$$\Rightarrow u^{(n+1)} = Au^{(n)} + \dots$$

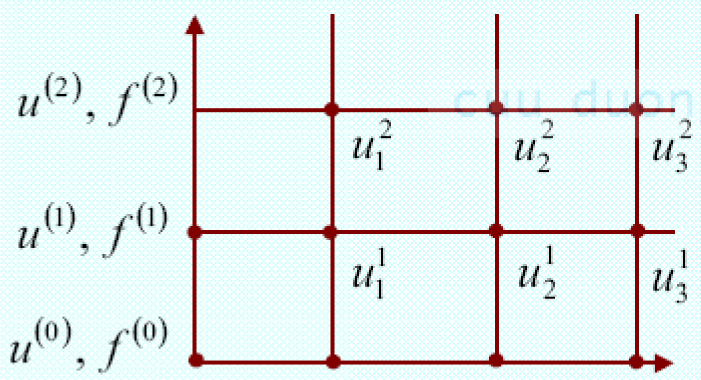
Sai phân lùi (sơ đồ ẩn)

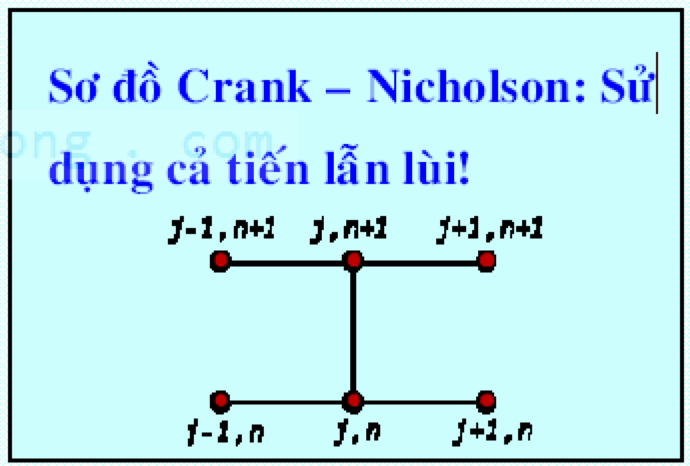


$$\Rightarrow Bu^{(n+1)} = u^{(n)} + \dots$$

BÀI TOÁN PARABOLIC (mô hình truyền nhiệt)

- Sơ đồ Crank-Nicholson





$$Au^{(n+1)} = Bu^{(n)} + \ldots$$

Kết quả: Chính xác hơn sơ đồ ẩn; Ổn định không điều kiện

Outline

- General introduction
  - Structure and operation of chemical engineering systems
  - What is a chemical process?
  - Motivation examples
- Part I: Process modeling
- Part II: Computer simulation
- Part III: Optimization of chemical processes