TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC

****

BÁO CÁO PHƯƠNG PHÁP SỐ

Phương pháp Runge – Kutta giải gần đúng phương trình vi phân thường

Giảng viên hướng dẫn: Hà Thị Ngọc Yến

Nhóm sinh viên thực hiện:

	Họ tên	MSSV
1.	Hoàng Phương Cúc	20185332
2.	Nguyễn Thị Quý	20185396

Hà Nội 01/2021

MỤC LỤC

Lời Mở Đầu

Trong lĩnh vực toán ứng dụng thường gặp rất nhiều bài toán liên hệ với phương trình vi phân thường. Việc nghiên cứu phương trình vi phân thuonwgff đóng vai trò quan trọng trong lý thuyết toán học. Một số trường hợp đơn giản, nghiệm của nó được tìm bằng phương pháp cầu phương còn nói chung không tìm được nghiệm chính xác.

Vì vậy, phương pháp gần đúng giúp các ngành kĩ thuật giải quyết một lớp khá lớn các bài toán dạng phương trình vi phân mà phương pháp cầu phương không cho phép. Hơn nữa, ngay cả các bài toán được giải bằng cầu phương dẫn đến nghiệm trong dạng hàm khá phức tạp cũng khó tính được giá trị của nghiệm tại một điểm nào đó. Phép tính gần đúng sẽ cho những thuật toán đơn giản mà đạt sự mong muốn phù hợp thực tế.

Trong bài báo cáo này, nhóm chúng em xin được trình bày về việc dùng phương pháp Runge Kutta để giải bài toán Cauchy và các phương trình vi phân thường khác.

1. CƠ SỞ LÝ THUYẾT

1.1. Phương trình vi phân thường

$$F(x, y, y', \ldots, y^{(n-1)}) = 0 \qquad (1.1)$$

- Trong đó *y = y(x)* là ẩn hàm cần tìm và nhất thiết phải có sự tham gia của đạo hàm (đến cấp nào đó) của ẩn.
- Trong trường hợp ẩn hàm cần tìm là hàm nhiều biến (xuất hiện các đạo hàm riêng) thì phương trình vi phân còn gọi là phương trình đạo hàm riêng. Để phân biệt người ta thường gọi phương trình với ẩn hàm là hàm một biến là phương trình vi phân thường là đối tượng chính được nói trong mục này.
- Thông thường ta xét các phương trình với ẩn hàm là hàm số một biến thực $y = y(x)$ xác định trên khoảng mở $I \subset R$, khi đó hàm F trong đẳng thức trên xác định trong một tập mở G của $R \ x \ R^{n+1}$.
- Trong trường hợp ẩn hàm cần tìm là vector hàm (hàm với giá trị vector) $y(x) = (y_1(x), \ldots, y_m(x))^T \in R^m$, F là một ánh xạ nhận giá trị trong R^m và (1.1) được hiểu là hệ phương trình vi phân. Ta nói một phương trình vi phân có cấp n nếu n là cấp lớn nhất của đạo hàm ẩn xuất hiện trong phương trình. Phương trình vi phân cấp I có dạng tổng quát $F(x, y, y') = 0$ trong đó $F(x, y, y') = 0$ được giả thiết là liên tục với các đạo hàm riêng của nó trên miền $G \subset R^3$. Với một số giả thiết thích hợp, phương trình vi phân thường cấp I có thể viết được dưới dạng sau (gọi là dạng giải ra đối với đạo hàm)

$$y' = f(x, y) \qquad (1.2)$$

với f liên tục trong một miền $D \subset R^2$.

Ví dụ: Các phương trình:

$$e^y + e^{y'} cosx = 1$$

$$(y''')^2 - 2xy = lnx$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial x^2} = 0$$

lần lượt là các phương trình vi phân thường cấp I, cấp III và phương trình đạo hàm riêng cấp II.

1.2. *Bài toán Cauchy*

Nghiệm của một phương trình vi phân nói chung phụ thuộc vào một hay nhiều hằng số tùy ý nào đó. Để xác định một nghiệm cụ thể, ta cần thêm một hay vài dữ kiện nào đó về nghiệm (tùy theo cấp của phương trình vi phân). Chẳng hạn $y = \frac{x^3}{3} + C$ là nghiệm tổng quát của phương trình $y' = x^2$. Dễ thấy:

$y = \frac{x^3}{3}$ + 1 là nghiệm duy nhất thỏa mãn $y(0) = 1$.

Phát biểu bài toán: Xét y(x) thỏa mãn

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases} \qquad (1.3)$$

Trong đó $(x_0, y_0) \in D$ là điều kiện ban đầu.

Chú ý: Không phải lúc nào bài toán Cauchy cũng có nghiệm, và khi có nghiệm cũng không nhất thiết có duy nhất nghiệm. Chẳng hạn:

Phương trình $y' = x^2$, $y(0) = 0$ có duy nhất một nghiệm là $y = \frac{x^3}{3}$

Phương trình $xy' = y$, $y(0) = 1$ không có nghiệm nào.

Phương trình $y' = y^{\frac{1}{3}}$, $y(0) = 0$ có ít nhất hai nghiệm là y ≡ 0 và $y^2 = \frac{8}{27}x^3$.

- ***Sự tồn tại và duy nhất nghiệm***

Định nghĩa: Cho hàm $f(x, y)$xác định trên miền D ⊂ R^2 ta nói hàm f thỏa mãn điều kiện Lipschitz theo biến y trên D nếu tồn tại số dương L (gọi là hằng số Lipschitz) sao cho:

$$|f(x, y_1) - f(x, y_2)| \leq L|\, y_1 - y_2| \text{ với mọi } (x, y_1), (x, y_2) \in D$$

Nhận xét: Điều kiện Lipschitz là yếu hơn so với điều kiện giới nội của đạo hàm riêng $\frac{\partial f}{\partial y}$ trên D. Thật vậy giả sử $\frac{\partial f}{\partial y}$ liên tục và $\left|\frac{\partial f}{\partial y}\right| \leq L$.

Khi đó áp dụng định lý Lagrange cho hàm f(x,y) theo biến y ta được:

$$f(x, y_1) - f(x,\ y_2) = (\ y_1 -\ y_2)\frac{\partial f}{\partial y}[x,\ y_1 + \theta(\ y_2\ -\ \ y_1)]$$

Định lý tồn tại và duy nhất nghiệm: Giả sử hàm số f(x,y) trong (1.3) liên tục và thỏa mãn điều kiện Lipschitz theo biến y trên hình chữ nhật

$$D = \{(x, y)\epsilon R^2 : |x - x_0\ | \leq a, |y - y_0\ | \leq b\}$$

Khi đó nghiệm của bài toán Cauchuy (1.3) là tồn tại và duy nhất trong đoạn

$$I \coloneqq [x_0 - h, x_0 + h], \text{ với } h \coloneqq \min\,(x, \frac{b}{M}) \text{ và } M = \max_{(x,y)\in D} |f(x, y)|.$$

Chứng minh:

Sự tồn tại: Chứng minh rằng phép lặp Picard hội tụ đều trên I đến một nghiệm của bài toán Cauchy.

Trước tiên ta chứng minh quy nạp rằng:

$$|y_{k+1}(x)\ -\ y_k(x)| \leq\ ML^k\frac{|x - x_0|^{k+1}}{(k+1)!}, \text{ với mọi x} \in \text{I}$$

với k=0, bất đẳng thức trên chính là, $\left|\int_{x_0}^{x} f(t, y_0(t)dt\right| \leq M|x\ -\ x_0|$ bất đẳng thức này đúng.

Giả sử ta có điều đó với k−1, khi đó với $x_0 - h \leq x \leq x_0 + h$ ta có:

$$|y_{k+1}(x) - y_k(x)|\ =\ \left|\int_{x_0}^{x}\left[f\big(t, y_k(t)\big) - f\big(t, y_{k-1}(t)\big)\right]dt\right|$$

$$\leq \int_{x_0}^{x}\left|f\big(t, y_k(t)\big) - f\big(t, y_{k-1}(t)\big)\right|dt$$

$$\leq L\int_{x_0}^{x}|y_k(t) - y_{k-1}(t)|dt$$

$$\leq ML^k \int_{x_0}^{x} \frac{|x - x_0|^k}{k!} dt = ML^k \frac{|x - x_0|^{k+1}}{(k+1)!}$$

(với $x_0 - h \leq x \leq x_0 + h$ ta đánh giá tương tự).

Xét dãy hàm trên I, ta có:

$$\left|y_{k+p}(x) - y_k(x)\right| \leq \left|y_{k+p}(x) - y_{k+p-1}(x)\right| + \left|y_{k+p-1}(x) - y_{k+p-2}(x)\right| + \cdots + |y_{k+1}(x) - y_k(x)| \leq \frac{M}{L}\left\{\frac{(L|x - x_0|)^{k+p}}{(k+p)!} + \ldots + \frac{(L|x - x_0|)^{k+1}}{(k+1)!}\right\}$$

$$\leq \frac{M}{L} \sum_{j \geq k+1} \frac{(Lh)^j}{j!}$$

Chuỗi số $\sum_{j=0}^{\infty} \frac{(Lh)^j}{j!}$ là hội tụ, nên phần dư của nó (xuất hiện trong biểu thức cuối cùng) có thể làm cho bé tùy ý khi k đủ lớn. Theo tiêu chuẩn Cauchy, dãy $\{y_k(x)\}$ hội tụ đều đến hàm y(x). Để chứng minh y(x) là nghiệm ta chỉ cần qua giới hạn trong đẳng thức

$$y_{k+1}(x) = y_0 + \int_{x_0}^{x} f(t, y_k(t))dt$$

Vì dãy hàm$\{y_k(x)\}$ hội tụ đều, f liên tục đều trên hình chữ nhật D nên dãy hàm $\{f(t, y_k(t)\}$ hội tụ đều trên I đến hàm $f(t, y(t))$. Do đó có thể chuyển giới hạn qua dấu tích phân để được đẳng thức $y(x) = y_0 + \int_{x_0}^{x} f(t, y_k(t))dt$.

Vậy $y(x)$ chính là nghiệm của bài toán Cauchy.

Tính duy nhất: Giả sử bài toán Cauchy (1.3) còn có nghiệm z(x), khi đó ta có $y(x) - z(x) = \int_{x_0}^{x} [f(t, y(t)) - f(t, z(t))]dt$

Suy ra $|y(x) - z(x)| = \left|\int_{x_0}^{x} [f(t, y(t)) - f(t, z(t))]dt\right| \leq 2M|x - x_0|$

Từ đó $|y(x) - z(x)| = \left|\int_{x_0}^{x} [f(t, y(t)) - f(t, z(t))]dt\right|$

$$\leq L\int_{x_0}^{x} |y(t) - z(t)|dt \leq 2ML^k \frac{|x - x_0|^2}{2}$$

Lặp lại quá trình trên, ta dễ dàng chứng minh được rằng với mọi số tự nhiên k :

$$|y(x) - z(x)| \leq 2ML^k \frac{|x - x_0|^{k+1}}{(k+1)!} \text{Với mọi } x \in I$$

Cho $k \to +\infty$ ta có $|y(x) - z(x)| = 0$ trên I. Như vậy một cách địa phương, y(x) là nghiệm duy nhất

1.3. Ý tưởng của phương pháp Runge – Kutta

Xét bài toán Cauchy đã nêu ở trên và thỏa mãn điều kiện Picard-Lindelof. Phương pháp số sẽ cho ta kết quả là một bảng các giá trị xấp xỉ của y(x) nên ta chia [x_0, X] thành n đoạn nhỏ bởi các điểm chia x_i.

Với $x_i = x_0 + ih$; $x_n = X$, $h = \frac{X - x_0}{n}$

Biết $y(x_0) = y_0$. Ta sẽ tính giá trị y_i là xấp xỉ của các $y(y_i)$. Dễ thấy nghiệm của bài toán tương đương với việc tìm nghiệm của bài toán tích phân:

$$y(x) = y_0 + \int_{x_0}^{x} f(t, y(t)dt \qquad (3.1)$$

Theo phép lặp Picard ở trên ta cũng có công thức tính xấp xỉ sau:

$$y(x + h) = y(x) + \int_{x}^{x+h} f(t, y(t)dt \quad (3.2)$$

Ta sẽ áp dụng các phương pháp cầu phương gần đúng để giải bài toán phương trình vi phân. Nội dung cơ bản của phương pháp cầu phương: Giả sử muốn tính $\int_a^b f(t)dt$ ta sẽ thay hàm f(t) bởi một đa thức nội suy φ(t). Do đó $\int_a^b f(t)dt$ sẽ được xấp xỉ bởi tích phân của đa thức. Mà như ta đã biết sẽ dễ dàng tính được tích phân của một hàm đa thức.

Giả sử ta có s mốc nội suy $t_1 \leq t_2 \leq \ldots t_s$ trên [a, b]. Sử dụng phương pháp nội suy bằng đa thức Lagrange ta được đa thức $\varphi(t)$ có bậc nhỏ hơn s thỏa mãn $f(t_1) = \varphi(t_1), f(t_2) = \varphi(t_2), \ldots$

$$\varphi(t) = \sum_{j=1}^{s} f(t_j)L_j(t) \; (j = 1, 2, \ldots, s)$$

Trong đó $L_j(t) = \prod_{i=1,i\neq j}^{s} \frac{(t-t_i)}{(t_j-t_i)}$. Khi đó:

$$\int_a^b f(t)dt \approx \sum_{j=1}^{s} A_j f(t_j)$$

Với A_j được tính theo công thức: $A_j = \int_a^b L_j(t)dt$. Việc tìm biểu thức hiện của các A_j có thể áp dụng các phương pháp tính gần đúng tích phân.

Ta sẽ xét một số trường hợp sau:

- Với s=1, $t_1 = a$: Ta có $\varphi(t) \equiv f(a)$. Do đó:

$$\int_a^b f(t)dt \approx (b-a)f(a)$$

  Áp dụng vào công thức (3.2) ta được: $y(x+h) = y(x) + hf(x, y(x))$

  Đây là công thức Euler dạng hiện với sai số đạt cấp O(h^2)

- Với s=1, $t_1 = b$: Tương tự như trên ta được:

$$\int_a^b f(t)dt \approx (b-a)f(b)$$

  Áp dụng công thức (3.2) ta có công thức Euler dạng ẩn:

$$y(x+h) = y + hf(x+h, y(x+h))$$

- Với s=1, $t_1 = \frac{(a+b)}{2}$: Tương tự ta có công thức tính gần đúng tích phân:

$$\int_a^b f(t)dt \approx (b-a)f\left(\frac{a+b}{2}\right)$$

  Do đó ta được:

$$\int_x^{x+h} f(t, y(t)dt \approx hf\left(x + \frac{h}{2}, y\left(x + \frac{h}{2}\right)\right)$$

  Suy ra công thức trung điểm:

$$y(x+h) \approx y(x) + hf\left(x + \frac{h}{2}, y\left(x + \frac{h}{2}\right)\right)$$

- Với s=2,$t_1 = a, t_2 = b$:

  Ta tìm được: $L_1(t) = \frac{t-b}{a-b}$, $L_2(t) = \frac{t-a}{a-b}$.

  Do đó ta có $A_1 = A_2 = \frac{b-a}{2}$. Khi đó ta được:

$$\int_a^b f(t)dt \approx \frac{b-a}{2}[f(a) + f(b)]$$

  Do đó ta được công thức hình thang ẩn cho (3.2):

$$y(x+h) \approx y(x) + \frac{h}{2}(f(x,y) + f(x+h, y(x+h))$$

- Với s=3, $t_1 = a$, $t_2 = \frac{a+b}{2}$, $t_3 = b$:

  Tương tự ta dễ dàng tìm được:

$$A_1 = \frac{A_2}{4} = A_3 = \frac{b-a}{6}.$$

  Ta được công thức:

$$\int_a^b f(t)dt \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

  Từ đó suy ra công thức dạng ẩn:

$$y(x+h) = y(x) + \frac{h}{6}\left[f(x,y) + 4f\left(x + \frac{h}{2}, y\left(x + \frac{h}{2}\right)\right) f(x\ ) + h, y(x+h))\right]$$

  Gọi các công thức ở trên là dạng ẩn bởi vì ở mỗi bước ta không tính được trực tiếp y(x + h) mà phải giải phương trình đại số đối với $y(x+h)$. Công thức dạng hiện là công thức ta có thể tính trực tiếp được $y(x+h)$ thông qua các giá trị trước đó $y(x)$ (ví dụ như công thức Euler dạng hiện đã nói ở trên)

1.4. Phương pháp Runge - Kutta

Bài toán Cauchy hay còn gọi là bài toán giá trị ban đầu: Tìm y(x) thỏa mãn điều kiện:

$$\begin{cases} y' = f(x,y) \quad x_0 \le x \le \bar{x} \\ y(x_0) = y_0 \end{cases} \qquad (1)$$

$$y = \sum_{i=0}^{\infty} \frac{{y_0}^{(i)}}{i!} h^i$$

Hai nhà toán học đức Runge và Kutta đề ra một phương pháp tìm y mà chỉ phải tính một hàm f(x,y) ở một số điểm khác nhau.

Đặt $y_1 = y_0 + \Delta y_0$, trong đó:

$$\Delta y_0 = p_{r1}k_1(h) + \dots + p_{rr}k_r(h)$$

Để thuận lợi cho việc tính đạo hàm sau này, ta sẽ sử dụng đổi biến

$$k_i(h) = hf(a_i , b_i)$$

Với $a_i = x_0 + \alpha_i h;\ \alpha_1 = 0\,;i = 1, 2, \dots, r$

$$b_i = y_0 + \beta_{i1}k_1(h) + \dots + \beta_{i,i-1}k_{i-1}(h)$$

Gọi $\varphi_r(h) := y(x_0 + h) - y_1 = y(x_0 + h) - y(x_0) - \Delta y_0$

Nếu $\varphi_r^{(s+1)}(0) \neq 0$ thì

$$\varphi_r(h) = \sum_{i=0}^{r} \frac{\varphi_r^{(i)}(0)}{i!} h^i + O(h^{s+1})$$

Runge-Kutta chọn các hệ số α_i , β_{ij}, p_{rj} từ điều kiện $\varphi_r^{(i)}(0) = 0$ với i = 0,1,...,s; $\varphi_r^{(s+1)}(0) \neq 0$ với s càng lớn càng tốt.

Như vậy

$$\varphi_r^{(i)}(0) = y_0^{(i)} - \left[p_{r1}k_1^{(i)}(0) + \dots + p_{rr}k_r^{(i)}(0) \right] = 0 \qquad (i = 0, 1, \dots, s),$$

Hay ta có hệ phương trình phi tuyến để xác định các hệ số α_i , β_{ij}, p_{rj} ta cần giải hệ phương trình phi tuyến

$$\begin{cases} p_{r1}k_1^{(i)}(0) + \dots + p_{rr}k_r^{(i)}(0) = y_0^{(i)} \\ \mathrm{i} = 0, 1, \dots, s \end{cases} \qquad (1)$$

- **Phương pháp Euler**

Trường hợp riêng của phương pháp Runge-Kutta khi r = 1.

Ta có $\Delta y_0 = p_{11}k_1(h);\ k_1(h) = hf(x_0,\ y_0)$

$$y_1 = y_0 + \Delta y_0 = y_0 + p_{11}k_1(h) = y_0 + p_{11}hf(x_0,\ y_0)$$

mà $\varphi_1(h) := y(x_0 + h) - y\, nên$:

$\varphi_1(h) = y(x_0 + h) - y_0 - p_{11}hy(x_0, y_0);\ \ \varphi_1(0) = 0;$

$\varphi_1'(0) = y_0' - p_{11}f(x_0, y_0) = f(x_0, y_0) - p_{11}f(x_0, y_0) = (1 - p_{11})f(x_0, y_0)$

Để $\varphi_1'(0) = 0$ với mọi hàm f, ta phải có $p_{11} = 1$. Nói chung $\varphi_1''(0) = y_0'' \neq 0$ vậy $\Delta y_0 = p_{11}k_1(h) = hf(x_0, y_0)$.

Ta nhận được ***công thức Euler:***

$$\boxed{y_1 = y_0 + hf(x_0, y_0) \qquad (2)}$$

Nói chung: $y_{n+1} = y_n + h.f(x_n, y_n)$

$x_n = x_0 + n.h$

- **Phương pháp Euler cải tiến (công thức RK2)**

Trong phương pháp Runge -Kutta (RK), ta xét trường hợp r = 2

$$\Delta y_0 = p_{21}k_1(h) + p_{22}k_2(h)$$

Phương trình (2) trong trường hợp này có dạng:

$$y_0^{(l)} = p_{21}k_1^{(l)}(h) + p_{22}k_2^{(l)}(h) \qquad (l = 1,2 \ldots) \qquad (3)$$

Từ hệ thức (3) ta suy ra:

$$\begin{cases} y_0' = f_0 = p_{21}f_0 + p_{22}f_0 \\ y_0'' = p_{21}k_1''(0) + p_{22}k_2''(0) = 2p_{22}\left(\frac{\partial f_0}{\partial x}\alpha_2 + \frac{\partial f_0}{\partial y}\beta_{21}f_0\right) \end{cases} \qquad (4)$$

Từ phương trình đầu của (4) suy ra $p_{21} + p_{22} = 1$

Biến đổi phương trình thứ hai của hệ (4) ta được

$$(1 - 2\alpha_2 p_{22})\frac{\partial f_0}{\partial x} + (1 - 2p_{22}\beta_{21})\frac{\partial f_0}{\partial y} = 0 \quad (5)$$

CT RK2 (ứng với $r = 2$) đúng cho mọi hàm f nên để (5) nghiệm đúng, cần $1 - 2\alpha_2 p_{22} = 1 - 2p_{22}\beta_{21} = 0$.

Như vậy $\alpha_2 = \beta_{21} = 1; p_{21} = p_{22} = \frac{1}{2}$ và

$$\Delta y_0 = \frac{1}{2}h\{f(x_0, y_0) + f\big(x_0 + h, y_0 + hf(x_0, y_0)\big)\}$$

Ta nhận được công thức RK2, còn gọi là ***<u>công thức Euler cải tiến.</u>***

$$\begin{cases} \bar{y}_0 = y_0 + hf(x_0, y_0) \\ y_1 = y_0 + \frac{1}{2}h[f(x_0, y_0) + f(x_1, \bar{y}_0)] \end{cases} \qquad (6)$$

$$\begin{cases} k_1(h) = hf(x_0, y_0) \\ k_2 = hf(x_0 + h, y_0 + k_1(h)) \\ y_1 = y_0 + \dfrac{1}{2}(k_1 + k_2) \end{cases}$$

- **Công thức RK3**

Khi $r = 3$

$$\Delta y_0 = p_{31}k_1(h) + p_{32}k_2(h) + p_{33}k_3(h)$$

Phương trình (1) trong trường hợp này có dạng:

$$y_0^{(l)} = p_{31}k_1^{(l)}(h) + p_{32}k_2^{(l)}(h) + p_{33}k_3^{(l)}(h) \qquad (l = 1,2 \ldots)$$

Trong đó:

$k_1(h) = h(x_0, y_0)$

$k_2(h) = hf(x_0 + \alpha_2 h, y_0 + \beta_{21}k_1(h)$

$k_3(h) = hf(x_0 + \ \alpha_3 h; y_0 + \beta_{31}k_1(h) + \beta_{32}k_2(h))$

Tương tự ở RK2, ta giải được hệ phương trình

$$\begin{cases} p_{31} + p_{32} + p_{33} = 1 \\ p_{32}\alpha_2 + p_{33}\alpha_3 = \dfrac{1}{2} \\ p_{32}\beta_{21} + p_{33}(\beta_{31} + \beta_{32}) = \dfrac{1}{2} \\ \dfrac{1}{2}(p_{32}\alpha_2^2 + p_{33}\alpha_3^2) = \dfrac{1}{6} \\ p_{22}\alpha_2^2 + p_{33}\alpha_3(\beta_{31} + \beta_{32}) = \dfrac{1}{3} \\ p_{32}\alpha_2^2 + p_{33}(\beta_{31}^2 + \beta_{31}\beta_{32} + \beta_{32}^2) = \dfrac{2}{3} \\ p_{33}\beta_{32}\beta_{21} = \dfrac{1}{6} \\ p_{33}\beta_{32}\alpha_2 = \dfrac{1}{6} \end{cases}$$

Giải hệ phương trình trên, ta được:

$$p_{31} = \frac{1}{6} = p_{33}; p_{32} = \frac{2}{3}; \alpha_2 = \frac{1}{2} = \beta_{21}; \alpha_3 = 1; \beta_{31} = -1; \beta_{32} = 2$$

Ta được **<u>công thức RK3:</u>**

$$\begin{cases} \Delta y_0 = \dfrac{1}{6}(k_1 + 4k_2 + k_3) \\ k_1(h) = hf(x_0, y_0) \\ k_2(h) = hf\left(x + \dfrac{h}{2}, y + \dfrac{k_1}{2}\right) \\ k_3(h) = hf(x_0 + h, y - k_1 + 2k_2) \end{cases} \quad (7)$$

- **Công thức RK4**

$$Với\ r = 4, \varphi_4(h) = y(x_0 + h) - y(x_0) - \sum_{i=1}^{4} p_{4i}k_i'(h)$$

Dễ thấy $\varphi_4(0) = 0; \varphi_4'(0) = y_0' - \sum_{i=1}^{4} p_{4i}k_i'(0) = (1 - \sum_{i=1}^{4} p_{4i})f_0 = 0$

Từ đây suy ra $\sum_{i=1}^{4} p_{4i} = 1$

Ngoài ra ta còn đòi hỏi $\varphi_4''(0) = \varphi_4'''(0) = \varphi_4^{(4)}(0) = 0$

Kết quả là hệ gồm 11 phương trình đối với 13 ẩn
$\alpha_2, \alpha_3, \alpha_4, \beta_{21}, \beta_{31}, \beta_{32}, \beta_{41}, \beta_{42}, \beta_{43}, p_{4i}$ $(i = 1,4)$

Công thức RK4 thông dụng nhất có dạng:

$$\begin{cases} \Delta y_0 = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = hf(x_0, y_0) \\ k_2 = hf(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}) \\ k_3 = hf(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}) \\ k_4 = hf(x_0 + h, y_0 + k_3) \end{cases} \quad (8)$$

- **Công thức RK5**

$(r = 5)$

Công thức RK5 thông dụng nhất có dạng:

$$\begin{cases} \Delta y_0 = y_0 + \frac{7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6}{90} \\ k_1 = hf(x_0, y_0) \\ k_2 = hf\left(x_0 + \frac{h}{4}, y_0 + \frac{k_1}{4}\right) \\ k_3 = hf\left(x_0 + \frac{h}{4}, y_0 + \frac{k_1}{8} + \frac{k_2}{8}\right) \\ k_4 = hf\left(x_0 + \frac{h}{2}, y_0 - \frac{k_2}{2} + k_3\right) \\ k_5 = hf\left(x_0 + \frac{3h}{4}, y_0 + \frac{3k_1}{16} + \frac{9k_4}{16}\right) \\ k_6 = hf(x_0 + h, y_0 - \frac{3k_1}{7} + \frac{2k_2}{7} + \frac{12k_3}{7} - \frac{12k_4}{7} + \frac{8k_5}{7}) \end{cases} \quad (9)$$

- **PT vi phân thường bậc cao**

Ý tưởng: Ta sẽ dùng công thức RK4 giải phương trình vi phân bậc cao để đạt mức sai số ổn định và tốt nhất $O(h^5)$

Xét PTVP bậc m:

$$y^{(m)}(x) = f\left(x, y, y', \dots, y^{(m-1)}\right) \qquad a \le x \le b$$

Với $y(a) = \alpha_1,\ y'(a) = \alpha_2, \dots y^{(m-1)}(a) = \alpha_m$

Đặt $y_1 = y,\ y_2 = \ y', \dots,\ y_m = \ y^{(m-1)}$

Ta đưa được về dạng hệ phương trình vi phân cấp 1: $\begin{cases} Y' = F(x,y) \\ Y(a) = A \end{cases}$

Với $Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}, F = \begin{bmatrix} y_2 \\ y_3 \\ \vdots \\ f(x, y_1, y_2, \dots, y_m) \end{bmatrix}, A = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}$

Ta xét PTVP bậc 2 : $y'' + a(x)y' + b(x)y = c(x),\ y(x_0) = y_0\ ,\ y'(x_0) = y_1$

Đặt $y' = z = f(x, y, z)$

$$z' = -a(x)z - b(x)y + c(x) = g(x, y, z)$$

Với điều kiện ban đầu: $z(x_0) = \ y_1\ ,\ y(x_0) = y_0$

Phương thức RK4 trở thành:

$$y_0 = \ y_0 + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$$

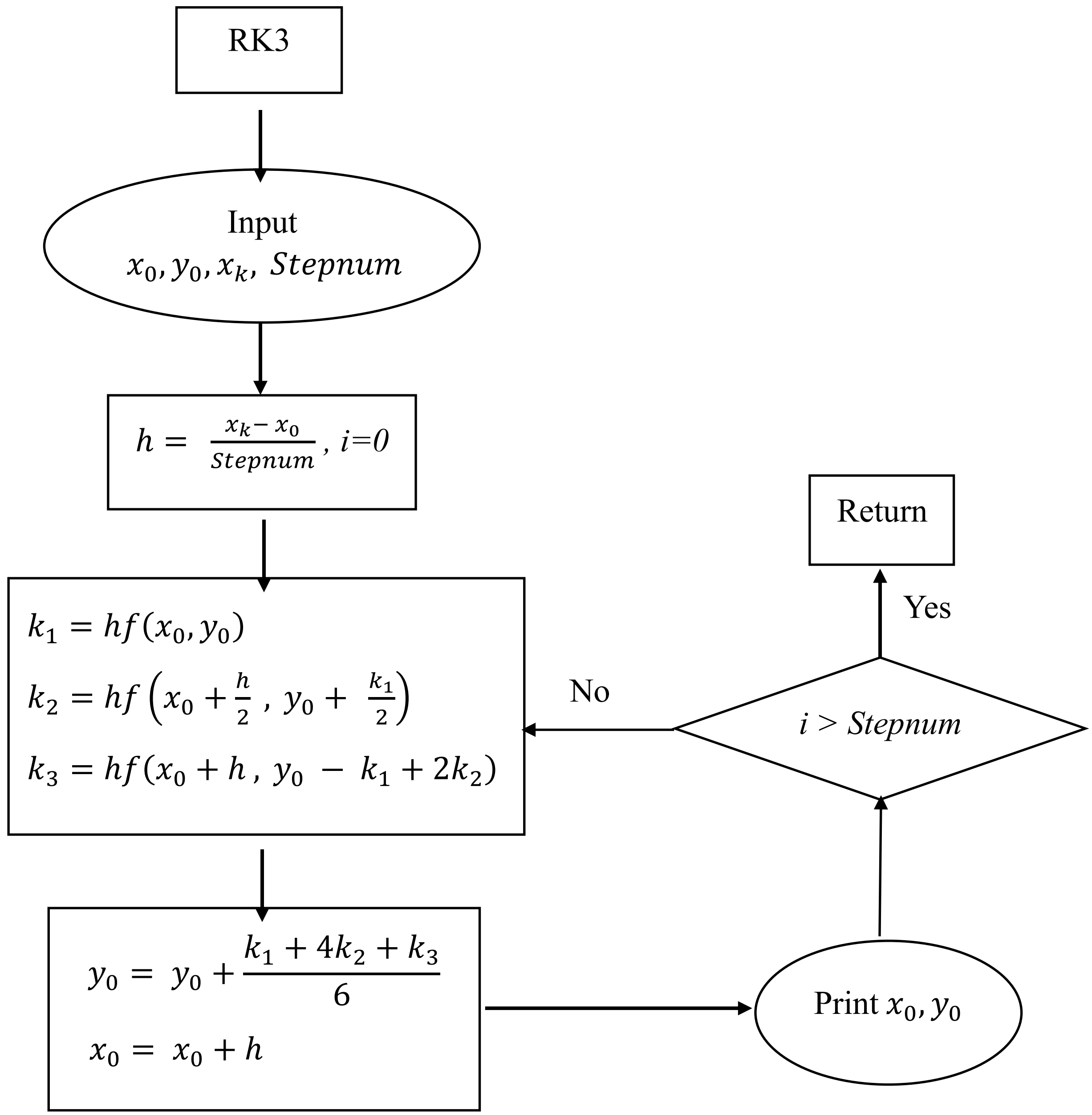
$$z_0 = \ z_0 + \frac{l_1 + 2l_2 + 2l_3 + l_4}{6}$$

Trong đó: $x_0 = \ x_0 + h$

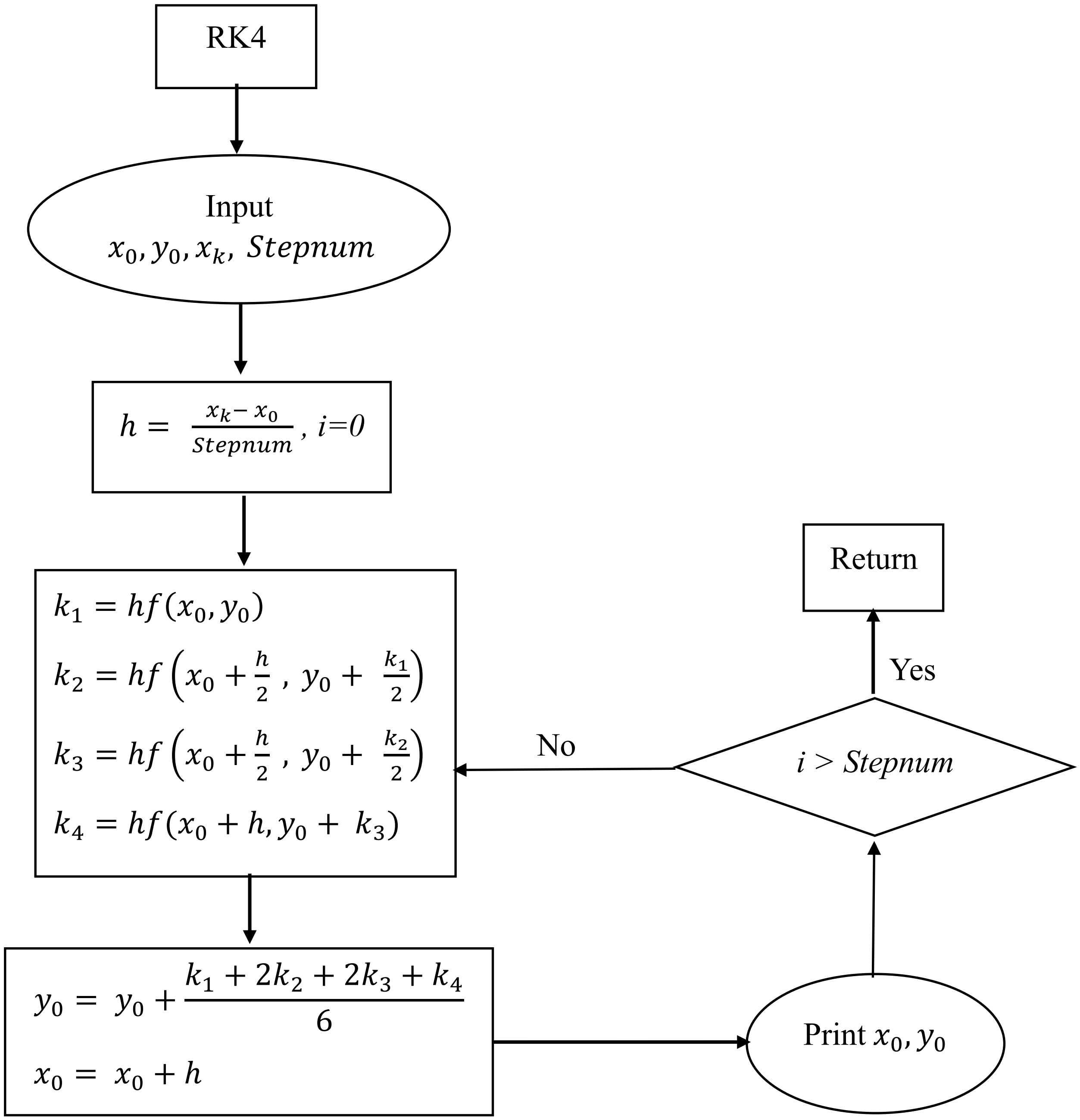
$$\begin{aligned}
k_1 &= hf(x_0, y_0, z_0) & l_1 &= hg(x_0, y_0, z_0) \\
k_2 &= hf\left(x_0 + \frac{h}{2}\ ,\ y_0 + \ \frac{k_1}{2}, z_0 + \frac{l_1}{2}\right) & l_2 &= hg\left(x_0 + \frac{h}{2}\ ,\ y_0 + \ \frac{k_1}{2}, z_0 + \frac{l_1}{2}\right) \\
k_3 &= hf\left(x_0 + \frac{h}{2}\ ,\ y_0 + \ \frac{k_2}{2}, z_0 + \frac{l_2}{2}\right) & l_3 &= hg\left(x_0 + \frac{h}{2}\ ,\ y_0 + \ \frac{k_2}{2}, z_0 + \frac{l_2}{2}\right) \\
k_4 &= hf(x_0 + h, y_0 + \ k_3, z_0 + \ l_3) & l_4 &= hg(x_0 + h, y_0 + \ k_3, z_0 + \ l_3)
\end{aligned} \tag{10}$$

2. THUẬT TOÁN

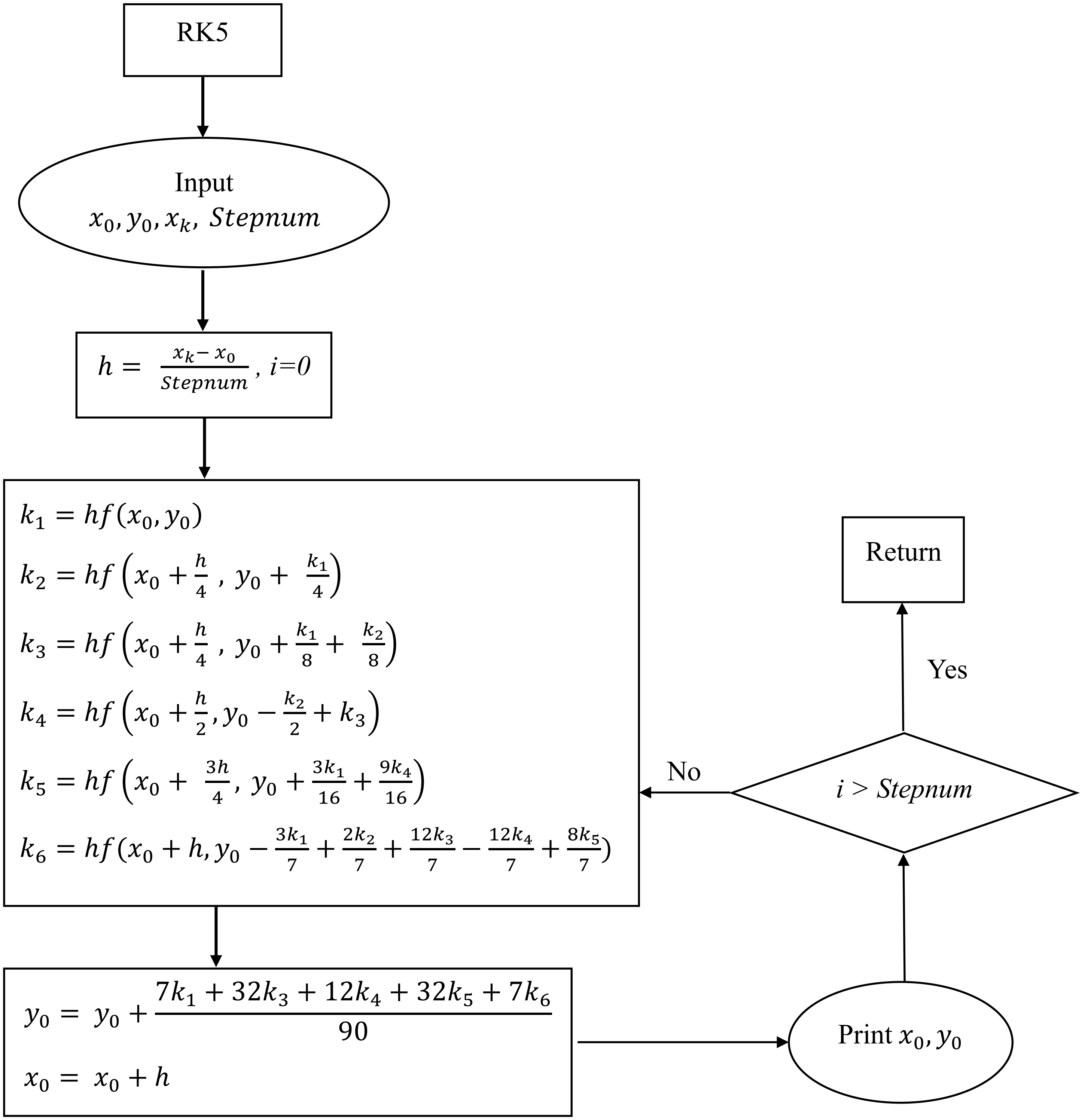
2.1. RK3



2.2. RK4

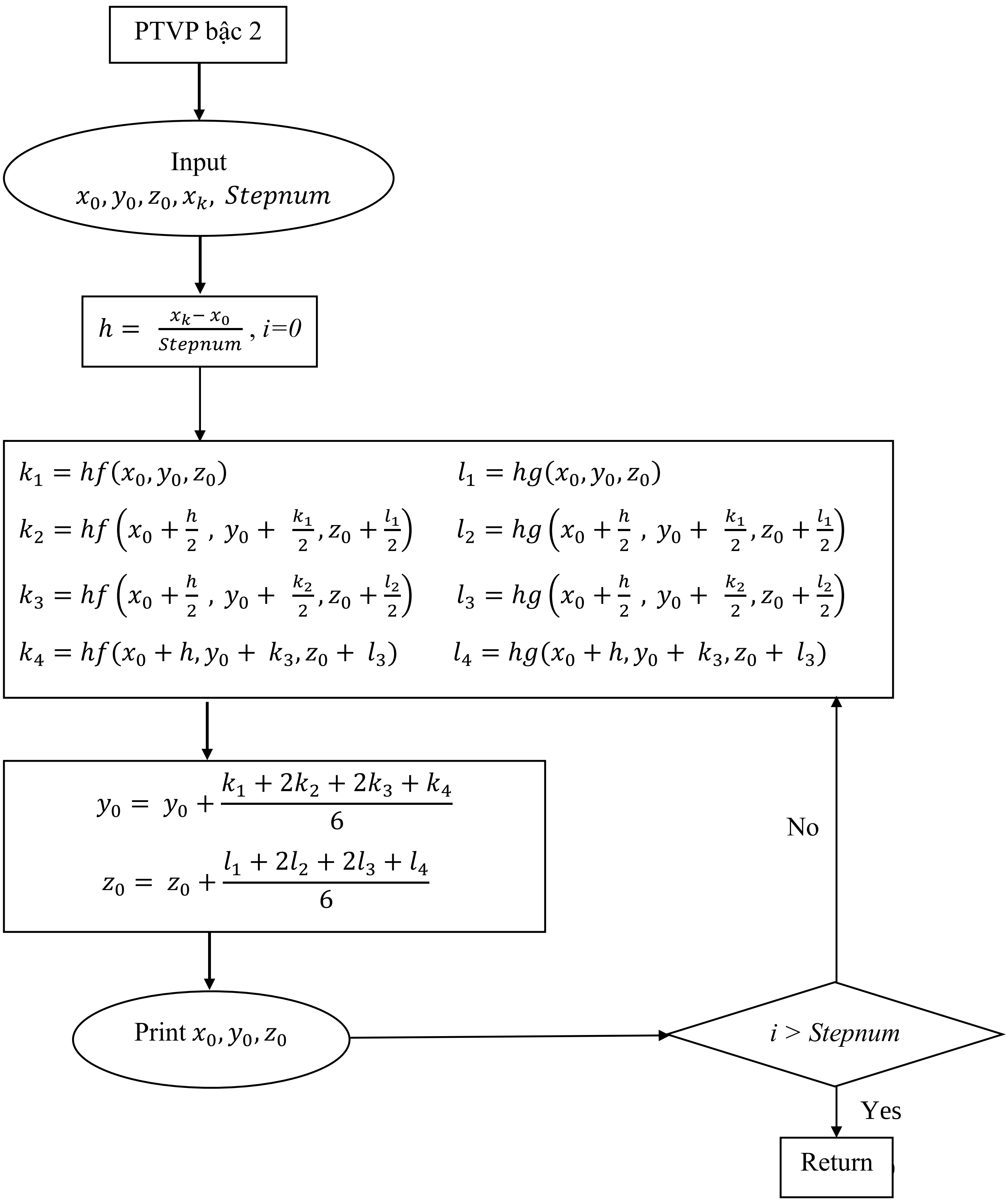

RK4
Input
$x_0, y_0, x_k, Stepnum$
$h = \frac{x_k - x_0}{Stepnum}$, $i=0$
$k_1 = hf(x_0, y_0)$
$k_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right)$
$k_3 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right)$
$k_4 = hf(x_0 + h, y_0 + k_3)$
$y_0 = y_0 + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$
$x_0 = x_0 + h$
Print x_0, y_0
$i > Stepnum$
No
Yes
Return

2.3. RK5



RK5
Input
$x_0, y_0, x_k, Stepnum$
$h = \frac{x_k - x_0}{Stepnum}$, $i=0$
$k_1 = hf(x_0, y_0)$
$k_2 = hf\left(x_0 + \frac{h}{4}, y_0 + \frac{k_1}{4}\right)$
$k_3 = hf\left(x_0 + \frac{h}{4}, y_0 + \frac{k_1}{8} + \frac{k_2}{8}\right)$
$k_4 = hf\left(x_0 + \frac{h}{2}, y_0 - \frac{k_2}{2} + k_3\right)$
$k_5 = hf\left(x_0 + \frac{3h}{4}, y_0 + \frac{3k_1}{16} + \frac{9k_4}{16}\right)$
$k_6 = hf(x_0 + h, y_0 - \frac{3k_1}{7} + \frac{2k_2}{7} + \frac{12k_3}{7} - \frac{12k_4}{7} + \frac{8k_5}{7})$
$y_0 = y_0 + \frac{7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6}{90}$
$x_0 = x_0 + h$
Print x_0, y_0
$i > Stepnum$
No
Yes
Return

2.4. PTVP bậc 2


PTVP bậc 2
Input
$x_0, y_0, z_0, x_k,\ Stepnum$
$h = \frac{x_k - x_0}{Stepnum}$, $i=0$
$k_1 = hf(x_0, y_0, z_0)$
$l_1 = hg(x_0, y_0, z_0)$
$k_2 = hf\left(x_0 + \frac{h}{2}\ ,\ y_0 + \ \frac{k_1}{2}, z_0 + \frac{l_1}{2}\right)$
$l_2 = hg\left(x_0 + \frac{h}{2}\ ,\ y_0 + \ \frac{k_1}{2}, z_0 + \frac{l_1}{2}\right)$
$k_3 = hf\left(x_0 + \frac{h}{2}\ ,\ y_0 + \ \frac{k_2}{2}, z_0 + \frac{l_2}{2}\right)$
$l_3 = hg\left(x_0 + \frac{h}{2}\ ,\ y_0 + \ \frac{k_2}{2}, z_0 + \frac{l_2}{2}\right)$
$k_4 = hf(x_0 + h, y_0 + \ k_3, z_0 + \ l_3)$
$l_4 = hg(x_0 + h, y_0 + \ k_3, z_0 + \ l_3)$
$y_0 = \ y_0 + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6}$
$z_0 = \ z_0 + \frac{l_1 + 2l_2 + 2l_3 + l_4}{6}$
Print x_0, y_0, z_0
i > $Stepnum$
No
Yes
Return

3. VÍ DỤ VÀ CHƯƠNG TRÌNH

VD1: Cho phương trình vi phân:

$$\begin{cases} y' = x + y \ (0 \le x \le y) \\ y(0) = 1 \end{cases}$$

a.Tìm nghiệm gần đúng với h = 0.1 bằng công thức RK3, RK4, RK5.

b.So sánh sai số với bước nhảy h = 0.1 với h = 0.05 bằng công thức RK4

Công thức nghiệm: $y = -x - 1 - 2e^x$

<u>Giải</u>

a)

Với h = 0.1 bằng công thức RK4:

$y_{i+1} = y_i + \frac{\Delta y_i}{6} \qquad i = 0,1,2,3,4$

Trong đó: $\Delta y_i = k_1^{(i)} + 2k_2^{(i)} + 2k_3^{(i)} + k_4^{(i)}$

Từ $y_0 = y(x_0) = 1$ và f(x,y) = x + y, ta có

$k_1^{(0)} = 0.1[0 + 1] = 0.1$

$k_2^{(0)} = 0.1[(0 + 0.05) + (1 + \frac{0.1}{2})] = 0.11$

$k_3^{(0)} = 0.1[(0 + 0.05) + (1 + \frac{0.11}{2})] = 0.1105$

$k_4^{(0)} = 0.1[(0 + 0.1) + (1 + 0.1105)] = 0.12105$

Vậy $\frac{1}{6}\Delta y_0 = \frac{1}{6}[0.1 + 2 \times 0.11 + 2 \times 0.1105 + 0.12105] = 0.11034166667$

và $y_1 = y(0{,}1) \approx y_0 + \frac{1}{6}\Delta y_0 = 1 + 0.11034166667 = 1.11034166667$

Tiếp tục xem y_1 đúng , tính $y_2 = y(0.2);\ y_3 = y(0.3); y_4 = y(0.4); y_5 = y(0.5)$

Tương tự ta có bảng kết quả:

x_k	y	y_k		
		RK3	**RK4**	**RK5**
0	1	1	1	1
0.1	1.1103418361512953	1.1103333333333334	1.1103416666666667	1.1103418364583333
0.2	1.2428055163203398	1.2427867222222222	1.242805141701389	1.2428055169989987
0.3	1.3997176151520063	1.399686459175926	1.3997169941250756	1.3997176162770575
0.4	1.5836493952825408	1.5836034851325942	1.5836484801613715	1.5836493969403727
0.5	1.7974425414002564	1.7973791183190386	1.7974412771936765	1.797442543690491

Sai số địa phương:

+ Với RK3: 10^{-4}
+ Với RK4: 10^{-5}
+ Với RK5: 10^{-8}

Code chương trình trên Python:

+ Input:
  - Hàm f(x,y) = x + y, nghiệm: $y = -x - 1 - 2e^x$
  - $x_0 = 0, y_0 = 1, x = 0.5 , stepnum = 5$
+ Output:
  - Nghiệm gần đúng y_k ứng với mỗi $x_k = x_0 + h * k$ $(k = \overline{0, stepnum - 1}, h = \frac{x - x_0}{stepnum})$
  - Đồ thị sai số
  - Đồ thị biểu diễn nghiệm
+ Code:
  Xử lí:

```
#vd1
def f(x, y):
    return x + y
#nghiem dung vd1: y = -x-1-2*e^(x)
def nghiem(x):
  return - x - 1 + 2*math.exp(x)
```

RK3:

```
def RK3(x0, y0, x, stepnum):
    h = (x-x0)/stepnum
    xk =[x0]
    yk =[y0]
    print("xk", "".ljust(2),"|","yk", "".ljust(25))
    print("----------------------------")
    print(x0,"".ljust(3),"|",y0,"".ljust(7))
    for i in range(0,stepnum):
        k1 = f(x0,y0)
        k2 = f(x0+h/2, y0+h*k1/2)
        k3 = f(x0+h,y0 - h*k1 +2*h*k2)
        y0 = y0 + h*(k1/6+2/3*k2+k3/6)
        x0 = x0 + h
        print(str(np.round(x0,3)).ljust(5),"|",str(y0).ljust(7))
        xk.append(x0)
        yk.append(y0)
    return np.asanyarray(xk), np.asanyarray(yk)
```

RK4:

```
def RK4(x0, y0, x, stepnum):
    h = (x-x0)/stepnum
    xk = [x0]
    yk = [y0]
    print("xk", "".ljust(2),"|", "yk", "".ljust(7))
    print("----------------------------")
    print(x0,"".ljust(3),"|", y0,"".ljust(8))
    for i in range(0, stepnum):
        k1 = f(x0, y0)
        k2 = f(x0 + h/2, y0 + h*k1/2)
        k3 = f(x0 +h/2, y0 + h*k2/2)
        k4 = f(x0 +h, y0 + h*k3)
        x0 = x0 + h
        y0 = y0 + h*(k1+2*k2+2*k3+k4)/6
        print(str(np.round(x0,3)).ljust(5) , "|", str(y0).ljust(7))
        xk.append(x0)
        yk.append(y0)
    return np.asarray(xk), np.asarray(yk)
```

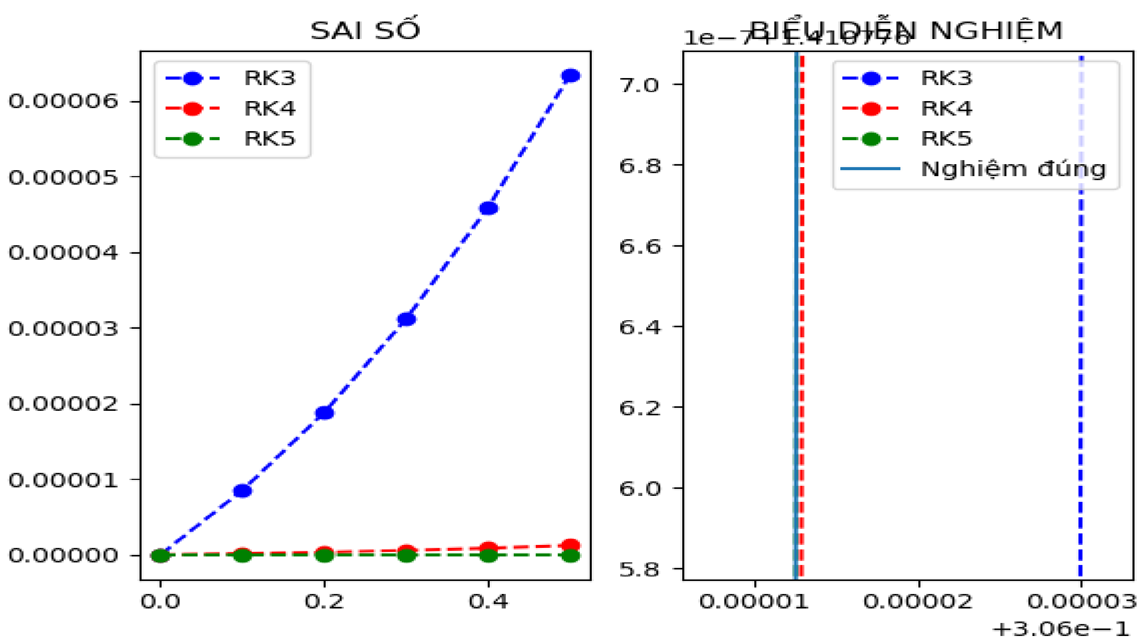
RK5:

```
def RK5(x0, y0, x, stepnum):
    h = (x-x0)/stepnum
    xk =[x0]
    yk =[y0]
    print("xk", "".ljust(2),"|", "yk", "".ljust(7))
    print("----------------------------")
    print(x0,"".ljust(3),"|", y0,"".ljust(8))
    for i in range(0, stepnum):
        k1 = f(x0, y0)
        k2 = f(x0 + h/4, y0 + h*k1/4)
        k3 = f(x0 + h/4, y0 + h*k1/8 + h*k2/8)
        k4 = f(x0 + h/2, y0 - h*k2/2 + h*k3)
        k5 = f(x0 +3*h/4, y0 +3/16*h*k1+9/16*h*k4)
        k6 = f(x0+ h, y0 - 3/7*h*k1 + 2/7*h*k2 + 12/7*h*k3 - 12/7*h*k4 + 8/7*h*k5)
        x0 = x0 + h
        y0 = y0 + h/90*( 7*k1 + 32*k3 +12*k4 + 32*k5 +7*k6)
        print(str(np.round(x0,3)).ljust(5) , "|", str(y0).ljust(7))
        xk.append(x0)
        yk.append(y0)
    return np.asarray(xk), np.asarray(yk)
```

Chạy chương trình: main(1)

Kết quả chạy chương trình:

```
RK3
xk     | yk
---------------------------
0      | 1
0.1    | 1.1103333333333334
0.2    | 1.2427867222222222
0.3    | 1.399686459175926
0.4    | 1.5836034851325942
0.5    | 1.7973791183190386

RK4
xk     | yk
---------------------------
0      | 1
0.1    | 1.1103416666666668
0.2    | 1.242805141701389
0.3    | 1.3997169941250756
0.4    | 1.5836484801613715
0.5    | 1.7974412771936765

RK5
xk     | yk
---------------------------
0      | 1
0.1    | 1.1103418364583333
0.2    | 1.2428055169989987
0.3    | 1.3997176162770575
0.4    | 1.5836493969403727
0.5    | 1.797442543690491

Nghiệm đúng [1.0, 1.1103418361512953, 1.2428055163203398, 1.3997176151520063, 1.5836493952825408, 1.7974425414002564]

Sai số RK3: 6.342308121776163e-05

Sai số RK4: 1.2642065798651458e-06

Sai số RK5: 2.2902346685782504e-09
```



b) Kết quả

x	y	y_k	
		h = 0.1	**h = 0.05**
0	1	1	1
0.05	1.0525421927520482		1.0525421875
0.1	1.1103418361512953	1.1103416666666668	1.1103418251086425
0.15	1.1736684854565662		1.1736684680433338
0.2	1.2428055163203398	1.242805141701389	1.2428054919123024
0.25	1.3180508333754828		1.318050801301153
0.3	1.3997176151520063	1.3997169941250756	1.399717574689427
0.35	1.4881350971865142		1.4881350475598514
0.4	1.5836493952825408	1.5836484801613715	1.5836493356584533
0.45	1.6866243709803375		1.686624300464123
0.5	1.797442541400256	1.7974412771936765	1.7974424590317473

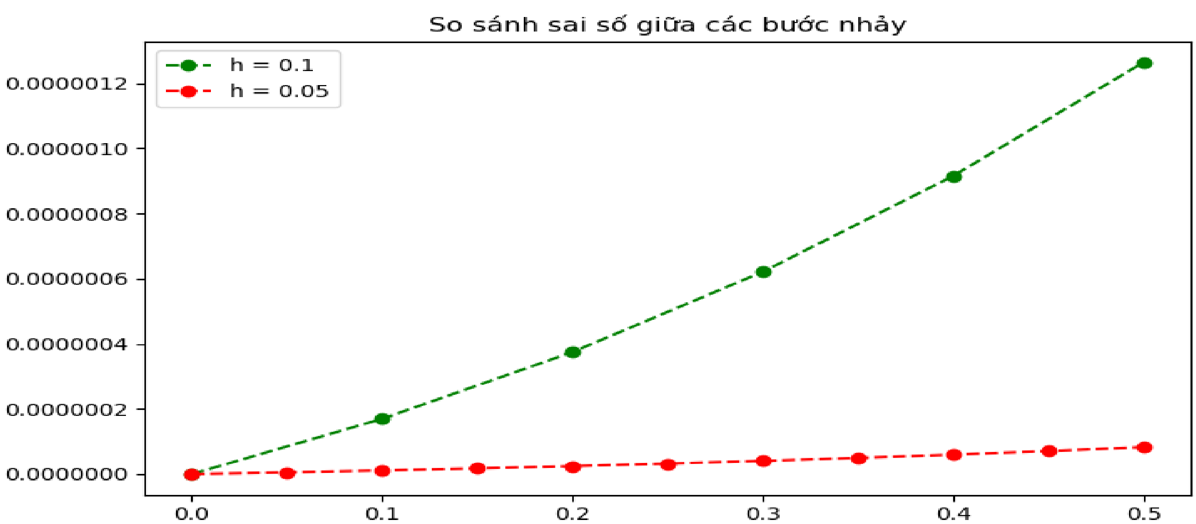
Sai số:

+ Với h = 0.1: 10^{-5}
+ Với h = 0.05: 10^{-7}

Chạy chương trình: main(2)

Kết quả:

```
h = 0.1
xk     | yk
----------------------------
0      | 1
0.1    | 1.1103416666666668
0.2    | 1.242805141701389
0.3    | 1.3997169941250756
0.4    | 1.5836484801613715
0.5    | 1.7974412771936765

h= 0.05
xk     | yk
----------------------------
0      | 1
0.05   | 1.0525421875
0.1    | 1.1103418251086425
0.15   | 1.1736684680433338
0.2    | 1.2428054919123024
0.25   | 1.318050801301153
0.3    | 1.399717574689427
0.35   | 1.4881350475598514
0.4    | 1.5836493356584533
0.45   | 1.686624300464123
0.5    | 1.7974424590317473
Sai số với h = 0.1:  1.2642065798651458e-06
Sai số với h = 0.05:  8.236850868037493e-08
```



Nhận xét:

+ Trên lí thuyết sai số của RK5 và RK4 là tương đồng nhau đạt O(h^5) địa phương nhưng khi chạy chương trình code nhận thấy sai số của RK5 rất tốt đạt mức O(h^8) địa phương
+ Bước nhảy càng nhỏ thì sai số đạt được càng tốt

VD2: Cho phương trình vi phân:

$$\begin{cases} y' = y - \dfrac{2x}{y} (0 \le x \le 0.5) \\ y(0) = 1 \end{cases}$$

Tìm nghiệm gần đúng với h = 0.1 bằng công thức RK3, RK4, RK5

Công thức nghiệm: $y = \sqrt{2x+1}$

Kết quả:

x_k	y	y_k		
		RK3	**RK4**	**RK5**
0	1	1	1	1
0.1	1.0954451150103321	1.0954445656918377	1.095445531693094	1.0954451127876526
0.2	1.1832159566199232	1.1832170026039872	1.1832167455059934	1.1832159530957684
0.3	1.2649110640673518	1.2649147918002834	1.2649122283403926	1.2649110592611132
0.4	1.3416407864998738	1.341647905494413	1.3416423537503719	1.3416407801952344
0.5	1.4142135623730951	1.4142246755924743	1.4142155778900853	1.414213554271966

Chương trình:

+ Input:
  - Hàm f(x,y) = $y - \frac{2x}{y}$, nghiệm: $y = \sqrt{2x+1}$
  - $x_0 = 0, y_0 = 1, x = 0.5, stepnum = 5$
+ Output:
  - Nghiệm gần đúng y_k ứng với mỗi $x_k = x_0 + h * k$ $(k = \overline{0, stepnum - 1}, h = \frac{x - x_0}{stepnum})$

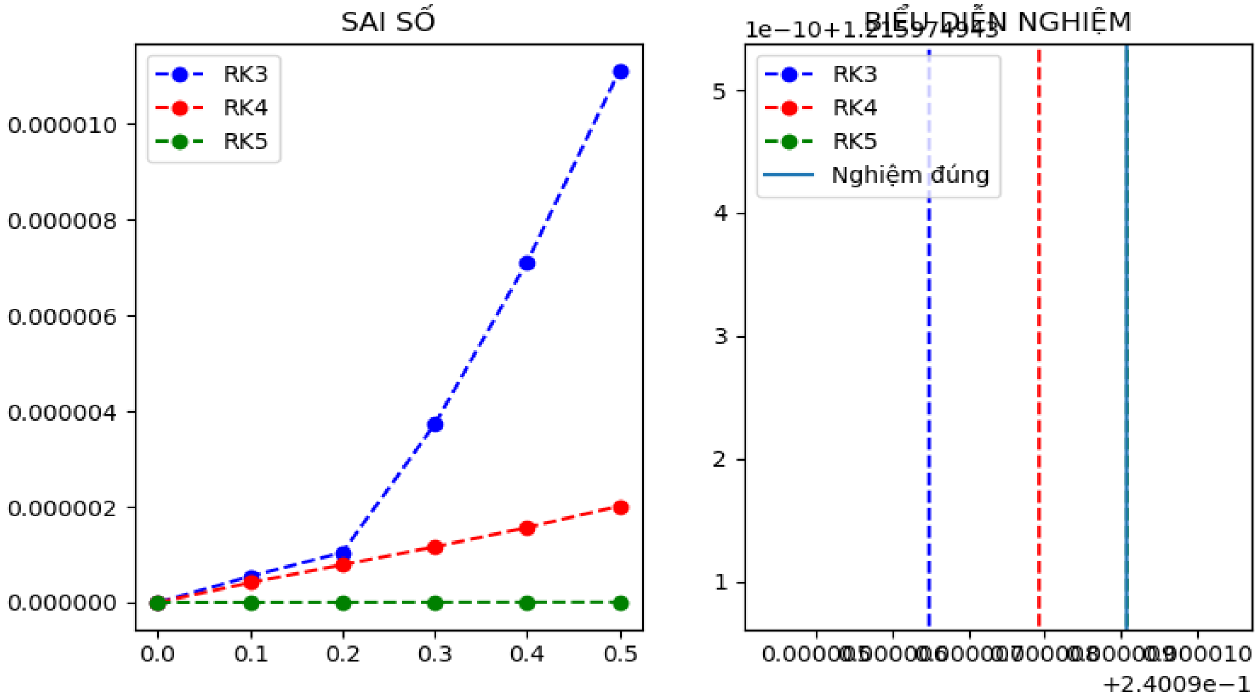
- Đồ thị sai số
- Đồ thị biểu diễn nghiệm

Xử lí:

```
#vd2
def f(x, y):
    return y - 2*x/y
#nghiem dung vd2: y = sqrt(2*x+1)
def nghiem(x):
    return math.sqrt(2*x + 1)
```

Chạy chương trình : main(1)

```
RK3
xk     | yk
----------------------------
0      | 1
0.1    | 1.0954445656918377
0.2    | 1.1832170026039872
0.3    | 1.2649147918002834
0.4    | 1.341647905494413
0.5    | 1.4142246755924743

RK4
xk     | yk
----------------------------
0      | 1
0.1    | 1.095445531693094
0.2    | 1.1832167455059934
0.3    | 1.2649122283403926
0.4    | 1.3416423537503719
0.5    | 1.4142155778900853

RK5
xk     | yk
----------------------------
0      | 1
0.1    | 1.0954451127876526
0.2    | 1.1832159530957684
0.3    | 1.2649110592611132
0.4    | 1.3416407801952344
0.5    | 1.414213554271966

Nghiệm đúng [1.0, 1.0954451150103321, 1.1832159566199232, 1.2649110640673518, 1.3416407864998738, 1.4142135623730951]

Sai số RK3: 1.1113219379188521e-05

Sai số RK4: 2.0155169901947545e-06

Sai số RK5: 8.101129100879234e-09
```



VD3: Tìm nghiệm của bài toán Cauchy cấp hai

$$\begin{cases} y'' + xy' + y = 0 \ (0 \le x \le 0.5) \\ y(0) = 0 \qquad\qquad y'(0) = 1 \end{cases}$$

Với h = 0.1

Giải:

Đặt $y' = z$

Bài toán đã cho tương đương với hệ phương trình vi phân cấp 1.

$$\begin{cases} y' = z & y(0) = 0 \\ z' = -xz - y & z(0) = 1 \end{cases}$$

f(x,y,z) = z

g(x,y,z) = -xz-y

Từ cặp giá trị $(y_0, z_0) = (0,1)$ ta sẽ tìm cặp giá trị (y_1, z_1) với bước h = 0.1. Áp dụng công thức (10) Kết quả tính được cho trong bảng sau:

i	x	y	z	k = hf	l = hg	Δy	Δz
0	0	0	1	$k_1 = 0.1$	0	$\Delta y_0 = k_1 + 2k_2 + 2k_3 + k_4 = 0.5980025$	$\Delta z_0 = l_1 + 2l_2 + 2l_3 + l_4 = -0.05935025$
	0.05	0.05	1	$k_2 = 0.1$	-0.01		
	0.05	0.05	0.995	$k_3 = 0.0995$	-0.009975		
	0.1	0.095	0.990025	$k_4 = 0.0990025$	-0.01940025		
$y_1 = y_0 + \frac{\Delta y_0}{6} = 0.099667083$ $z_1 = z_0 + \frac{\Delta z_0}{6} = 0.99010829$							
1	0.2	$y_2 = 0.19735406281023912$ $z_2 = 0.9605291874379521$					
2	0.3	$y_3 = 0.29115925558278766$ $z_3 = 0.9126522233251636$					
3	0.4	$y_4 = 0.37933316911719817$ $z_4 = 0.8482667323531206$					
4	0.5	$y_5 = 0.4603433544350426$ $z_5 = 0.7698283227824785$					

Nghiệm gần đúng của phương trình đã cho trên [0; 0.5]

x	0	0.1	0.2	0.3	0.4	0.5
y(x)	0	0.099667083	0.19735406281023	0.29115925558278766	0.37933316911719817	0.4603433544350426

- Phương trình vi phân cấp 2 với hệ số không hằng rất khó khăn trong việc giải ra nghiệm đúng, chúng em cũng chưa tìm ra được nghiệm đúng của bài toán đã cho. Tuy nhiên trong quá trình tìm hiểu chúng em thấy trong thư viện Scipy trong Python có hỗ trợ tìm nghiệm của phương trình vi phân cấp 2 nhưng chưa kiểm tra được sự đúng đắn của thuật toán.

- Khi so sánh sai số giữa nghiệm gần đúng bằng RK4 và nghiệm khi chạy trong thư viện chúng em nhận thấy sai số rất tốt và đạt sai số 10^{-7}

Chương trình:

+ Input:
    - f(x,y,z) = z; g(x,y,z) = -xz – y
    - $x_0 = 0, y_0 = 0, z_0 = 1, x = 0.5, stepnum = 5$
+ Output:
    - Nghiệm gần đúng y_k ứng với mỗi $x_k = x_0 + h * k$

$(k = \overline{0, stepnum - 1}, h = \frac{x - x_0}{stepnum})$

Code:

Xử lí:

```
#giai ptvp y" + xy' + y = 0
def f(x,y,z):
    return z
def g(x,y,z):
    return -x*z-y
```

RK4:

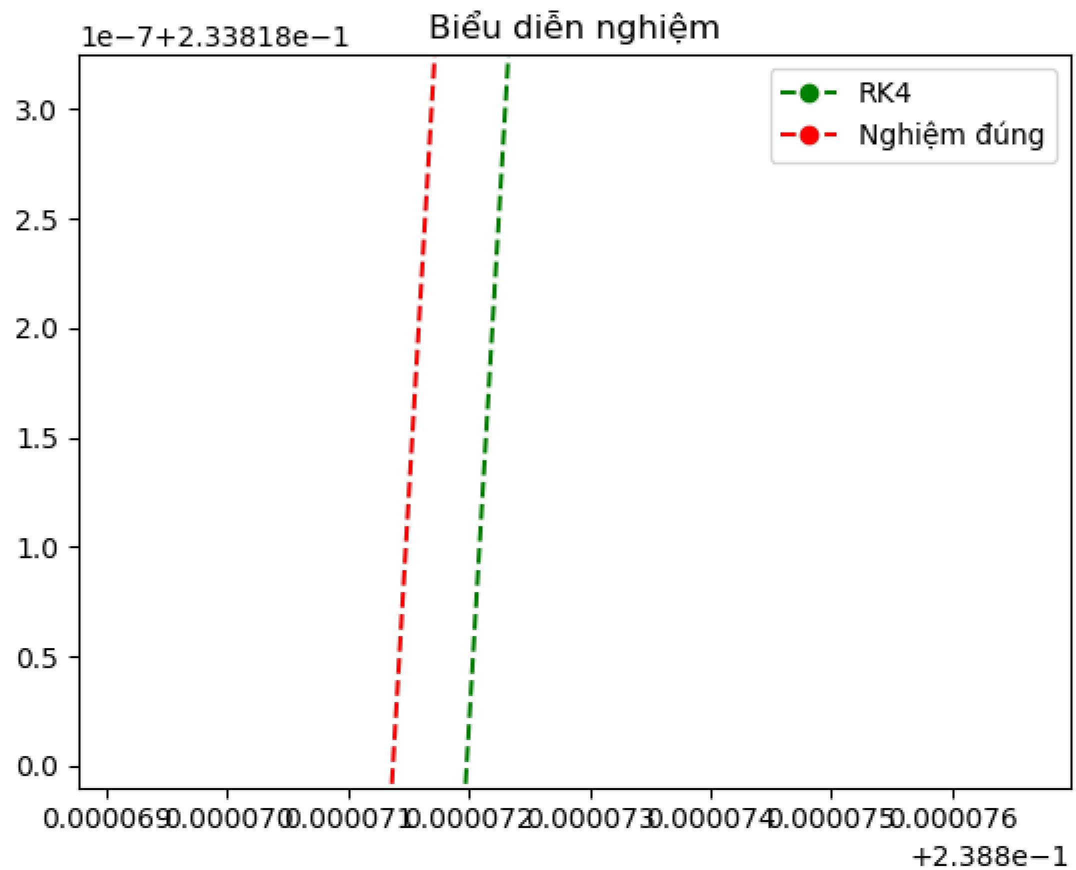
```
def RK4(x0, y0, z0, x, stepnum):
    h = (x-x0)/stepnum
    xk =[x0]
    yk =[y0]
    zk =[z0]
    print("xk", "".ljust(2),"|","yk", "".ljust(22), "|","zk".ljust(25))
    print("-----------------------------------------------------")
    print(x0,"".ljust(3),"|", y0,"".ljust(23),"|",z0,"".ljust(25))
    for i in range(0, stepnum):
        k1 = h*f(x0, y0 ,z0)
        l1 = h*g(x0, y0, z0)
        k2 = h*f(x0 + h/2, y0 + k1/2,z0 +l1/2)
        l2 = h*g(x0 + h/2, y0 + k1/2, z0 + l1/2)
        k3 = h*f(x0 + h/2, y0 + k2/2, z0 + l2/2)
        l3 = h*g(x0 + h/2, y0 + k2/2, z0 + l2/2)
        k4 = h*f(x0 + h, y0 + k3, z0 + l3)
        l4 = h*g(x0 + h, y0 + k3, z0 + l3 )
        y0 = y0 + 1/6*(k1 + 2*k2 + 2*k3 + k4)
        z0 = z0 + 1/6*(l1 + 2*l2 + 2*l3 + l4)
        x0 = x0 + h
        print(str(np.round(x0,3)).ljust(5) , "|", str(y0).ljust(25),"|",str(z0).ljust(7))
        xk.append(x0)
        yk.append(y0)
        zk.append(z0)
    return np.asarray(xk), np.asarray(yk), np.asarray(zk)
```

Hàm nghiệm sẵn trong Python:

```
#hàm giải sẵn ptvp cấp 2 trong thư viện scipy của python
import scipy as sp
from scipy.integrate import odeint
def h(y,x):
    y0 = y[0]
    y1 = y[1]
    y2 = -x*y1 - y0
    return y1, y2
#khoi tao y, y' tai x = 0
init = 0, 1
x = np.linspace(0,0.5,6)
sol = odeint(h, init, x)
kq = sol[:,0]
print('\nNghiệm tìm theo hàm trong thư viện của Pyhton')
print(sol)
```

Kết quả chạy:

```
xk     | yk                           | zk
-----------------------------------------------------
0      | 0                            | 1
0.1    | 0.09966708333333335          | 0.9900332916666666
0.2    | 0.19735406281023912          | 0.9605291874379521
0.3    | 0.29115925558278766          | 0.9126522233251636
0.4    | 0.37933316911719817          | 0.8482667323531206
0.5    | 0.4603433544350426           | 0.7698283227824785

Nghiệm tìm theo hàm trong thư viện của Pyhton
[[0.         1.        ]
 [0.09966734 0.99003327]
 [0.19735456 0.96052908]
 [0.29115994 0.91265201]
 [0.379334   0.84826638]
 [0.46034428 0.76982785]]

Sai số:
0.0
2.5902252905052325e-07
4.989981010061761e-07
6.885052059035779e-07
8.357653102919471e-07
9.258377692833442e-07
```



4. Sai số, nhận xét

❖ Sai số:

Sai số địa phương: Xét sai số mắc phải trên một bước với giả thiết bước trước nó tính đúng. Tại bước thứ i ta xét hàm $\bar{y}(x)$ là nghiệm của bài toán

$$\begin{cases} \bar{y}' = f(x, \bar{y}(x)) \\ \bar{y}(x_i) = y_i \end{cases}$$

Từ đó suy ra

$$\bar{y}(x_{i+1}) - y_{i+1} = \Delta y$$

Công thức RK3 đạt sai số O(h^4) địa phương và O(h^3) toàn cục

Công thức RK4 đạt sai số O(h^5) địa phương và O(h^4) toàn cục

❖ Nhận xét:

Bước nhảy càng nhỏ thì sai số đạt được càng tốt

Qua mỗi bước tính sai số lại tích lũy thêm. Nếu tính nghiệm gần đúng tại điểm quá xa so với mốc xuất phát x_0 sẽ có thể dẫn đến hiện tượng "sai một ly đi một dặm". Quá trình như vậy gọi là không ổn định.

Khối lượng tính toán còn lớn

KẾT LUẬN

Từ phương pháp Runge Kutta nhóm vừa trình bày ở trên, ta thấy qua mỗi bước tính thì sai số lại tích lũy thêm. Nếu tính nghiệm gần đúng tại điểm quá xa so với mốc xuất phát x_0 sẽ có thể dẫn đến hiện tượng "sai một ly đi một dặm". Quá trình như vậy gọi là không ổn định.

Đồng thời để tính y_{i+1} thì ta chỉ cần biết y_i , phương pháp như vậy gọi là phương pháp một bước. Vì vậy, các phương pháp trên chỉ được dùng để tính vài giá trị ở các mốc gần x_0.

Ngoài ra công thức dạng Rungge Kutta bậc 4, như đã tính trong thí dụ, mỗi bước tính cần đến bốn giá trị của hàm $f(x, y)$ tại các điểm khác nhau, khối lượng tính vẫn còn lớn.

TÀI LIỆU THAM KHẢO

1. Giáo trình giải tích số- Lê Trọng Vinh
2. Giải tích số - Phạm Kì Anh.
3. Phương pháp Runge-Kutta – Wikipedia
4. Tham khảo Báo cáo giải tích số - Tính gần đúng đạo hàm, tích phân
5. http://article.sapub.org/10.5923.j.ajcam.20170705.02.html