TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC

*****

BÁO CÁO GIẢI TÍCH SỐ

CHỦ ĐỀ 29: CÁC CÔNG THỨC RUNGE-KUTTA

Giảng viên hướng dẫn:

TS. Hà Thị Ngọc Yến

Sinh viên thực hiện:

Nguyễn Bảo Anh – MSSV: 20206110

Hà Nội, tháng 12/2021

MỤC LỤC

LỜI MỞ ĐẦU

Trong thực tế, hầu hết các hiện tượng tự nhiên đáng quan tâm đều liên quan tới biến đổi và thường được mô tả bởi các phương trình có liên quan đến sự thay đổi lượng, đó là phương trình vi phân. Nhiều bài toán khoa học kỹ thuật quy về việc tìm nghiệm phương trình vi phân thỏa mãn một số điều kiện nào đó (điều kiện ban đầu, điều kiện biên v.v...). Những phương pháp giải đúng chỉ áp dụng được cho một lớp hẹp của phương trình vi phân. Đa số các phương trình vi phân mô tả những hệ cơ học, lý học, hóa học, sinh học, nhìn chung rất phức tạp, không có hy vọng giải đúng. Vì vậy mà ta cần nghiên cứu phương pháp giải gần đúng. Trong bài báo cáo này, ta sẽ tìm hiểu về các công thức Runge-Kutta (gọi tắt là R-K) để giải gần đúng bài toán giá trị ban đầu của một phương trình vi phân thường, chẳng hạn như bài toán tăng trưởng tự nhiên tính toán dân số thế giới trong tương lai biểu diễn bằng bài toán Cauchy.

Để hoàn thành được bài báo cáo về các công thức R-K này rất cảm ơn sự hướng dẫn từ TS. Hà Thị Ngọc Yến. Tuy nhiên, do trình độ hiểu biết còn hạn chế và trong quá trình làm báo cáo không thể tránh khỏi những sai sót cho nên nếu mọi người có phát hiện ra sai sót hoặc có ý kiến muốn trao đổi có thể liên hệ qua địa chỉ thư điện tử hoặc bất kỳ hình thức nào khác. (Email: Anh.NB206110@sis.hust.edu.vn).

Xin cảm ơn!

Nguyễn Bảo Anh

A. KIẾN THỨC CHUNG

1. Phương trình vi phân.

a) Khái niệm

Phương trình vi phân là một phương trình toán học nhằm biểu diễn mối quan hệ giữa một hàm chưa được biết (một hoặc nhiều biến) với đạo hàm của nó (có bậc khác nhau).

Cấp của phương trình vi phân là bậc của đạo hàm có bậc cao nhất có trong phương trình.

b) Các loại phương trình vi phân

Có thể chia thành vài loại phương trình vi phân khác nhau, ta có thể chia phương trình vi phần thành 5 dạng chính:

- Phương trình vi phân thường (Ordinary Differential Equations - ODE) là phương trình vi phân trong đó hàm chưa biết là hàm 1 biến độc lập.
- Phương trình vi phân riêng phần (Partial Differential Equations - PDE) là phương trình vi phân trong đó hàm chưa biết là hàm của nhiều biến độc lập và các đạo hàm riêng của nó.
- Phương trình vi phân trì hoãn (Delay Differential Equation - DDE) là phương trình vi phân trong đó giá trị đạo hàm của hàm chưa biết tại một thời điểm nào đó là tính theo giá trị của hàm tại một thời điểm khác.
- Phương trình vi phân ngẫu nhiên (Stochastic differential equation - SDE) là phương trình vi phân trong đó một hoặc vài số hạng là quá trình ngẫu nhiên, vì thế dẫn đến hàm nghiệm cũng là một quá trình ngẫu nhiên (stochastic process).
- Phương trình vi phân đại số (Differential Algebraic Equation - DAE) là phương trình vi phân trong đó có chứa các số hạng là đại số và sai phân.

Mỗi loại trên lại chia thành tuyến tính và phi tuyến tính. Một phương trình vi phân là *tuyến tính* nếu mọi đạo hàm của nó đều tính theo hàm có dạng mũ 1 và không có tích hay hàm của các biến phụ thuộc.

2. Bài toán Cauchy cho phương trình vi phân thường

Như ở trên đã đề cập, phương trình vi phân thường (ODE) là phương trình vi phân mà trong đó hàm chưa biết là hàm một biến độc lập.

- Phương trình vi phân thường cấp 1 (ODE cấp 1) là phương trình vi phân có dạng: $y' = f(x, y)$. Trong đó y là hàm vector xác định trên đoạn $I = [a, b]$ và y nhận giá trị trên không gian $\mathbb{R}^m$.

- Tương tự như ODE cấp 1 thì ODE cấp k là phương trình vi phân có dạng: $y^{(k)} = f\left(x, y, y', \dots, y^{(k-1)}\right)$ trong đó x là biến số, y là hàm, $y^{(i)}$ là đạo hàm cấp i của y.

*** Bài toán điều kiện ban đầu (IVP-Initial Value Problem):**

Bài toán tìm nghiệm của các phương trình vi phân thường được chia làm 2 loại: bài toán giá trị ban đầu (IVP - Initial Value Problem) và bài toán điều kiện biên (BVP - Boundary Value Problem) phụ thuộc vào việc ta cần tìm nghiệm thỏa mãn điều kiện ban đầu hay điều kiện biên. Ở trong bài báo cáo này ta chỉ nói về bài toán điều kiện ban đầu (IVP).

Đa số các bài toán giá trị ban đầu mô tả các hệ thống được xét phụ thuộc thời gian và lời giải của bài toán phụ thuộc vào điều kiện tại thời điểm ban đầu.

- Đối với phương trình cấp 1, điều kiện ban đầu là $y(a) = y_0$

- Đối với phương trình cấp k, điều kiện ban đầu sẽ là:

$$y(a) = y_{00}, y'(a) = y_{01}, \dots, y^{(k-1)}(a) = y_{0,k-1}$$

Bài toán Cauchy là bài toán giải phương trình vi phân với điều kiện ban đầu và có thể được viết như sau:

Tìm hàm y(x) thỏa mãn điều kiện:

$$\begin{cases} y'(x) = f\left(x, y\right) & (a \le x \le b) \\ y(a) = y_0 & \left(y \in C^1([a, b], \mathbb{R}^n)\right) \end{cases}$$

VD1: Phương trình tăng trưởng: $\frac{dy}{dx} = ky$

Bài toán: Dân số thế giới vào giữa năm 1999 là 6 tỉ người và tăng thêm khoảng 212 nghìn người mỗi ngày. Giả sử mức tăng dân số tự nhiên tiếp tục với tỷ lệ này, hỏi vào năm 2020 dân số thế giới sẽ là bao nhiêu?

Ở đây ta sẽ có bài toán $\begin{cases} y'(x) = ky \quad (1998 \leq x \leq 2021) \\ y(1999) = 6 * 10^9 \end{cases}$ với k là hệ số tăng trưởng, $y(x)$ là hàm số biểu diễn dân số theo năm x. Ta có thể tìm ra hàm $y(x)$ để dự đoán ra dân số thế giới năm 2020 (khi $x = 2020$)

VD2: Phương trình về mô hình thú săn mồi (Phương trình Lotka-Volterra) với con mồi là thỏ và thú săn là cáo. $F(t)$ là hàm biểu diễn số lượng cáo và $R(t)$ là hàm biểu diễn số lượng thỏ tại thời điểm t, ta có:

$$\begin{cases} R'(t) = rR(1 - \frac{R}{K}) - aFR \\ F'(t) = aFR - dF \end{cases}$$ với r, K, b, d là các hằng số dương

Cụ thể có thể lấy $r = 0.6, K = 100, a = 0.4, d = 1.2$

Phương trình này khó có thể giải bằng phương pháp giải tích thông thường, chính vì vậy mà ta cần đến phương pháp giải số.

Nhận xét: Trong không gian nhiều chiều xét một phương trình vi phân cấp k, ta luôn đưa được về bài toán Cauchy cho phương trình vi phân cấp 1 bằng cách đổi biến hàm.

Đặt $u = \begin{bmatrix} y \\ y' \\ \vdots \\ y^{(k-1)} \end{bmatrix}$ và $u_1 = y, u_2 = y', \ldots, u_k = y^{(k-1)}$

$$\Rightarrow u' = \begin{bmatrix} y' \\ y'' \\ \vdots \\ y^{(k)} \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ f(x, u_1, u_2, \ldots, u_k) \end{bmatrix} = F(x, u), \; u(a) = \begin{bmatrix} y_{00} \\ y_{01} \\ \vdots \\ y_{0,k-1} \end{bmatrix}$$

Như vậy, ta có thể thấy rằng một bài toán phương trình vi phân thường cấp n trên không gian một chiều hoàn toàn có thể quy về bài toán phương trình vi phân cấp 1 trên không gian n chiều, hay làm việc trên vector n chiều. Từ đây, ta có thể coi mọi bài toán cấp cao hơn 1 quy về bài toán cấp 1, và ta sẽ chỉ làm việc với phương trình vi phân thường cấp 1.

Ta có một định lý như sau về nghiệm của bài toán Cauchy.

Định lý Picard–Lindelöf : Cho ODE cấp 1: $y' = f(x,y)$. Nếu hàm $f(x,y)$ liên tục Lipschitz theo biến y thì bài toán Cauchy luôn có nghiệm duy nhất với mọi giá trị ban đầu. Nghiệm của bài toán phụ thuộc liên tục theo giá trị ban đầu.

3. Khai triển Taylor

Xét một hàm số $f(x)$ là một hàm giải tích tại điểm a (tức là hàm có thể phân tích được thành dạng chuỗi lũy thừa của $(x-a)$ – xét trong không gian thực, nghĩa là hàm đó khả vi mọi cấp tại a). Khai triển Taylor của hàm $f(x)$ tại điểm a có dạng như sau:

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots = \sum_{i=0}^{\infty} \frac{f^{(i)}(a)}{i!}(x-a)^i$$

Ta cũng có thể khai triển tới cấp m bất kỳ mà ta mong muốn (tức là khai triển đến khi xuất hiện vô cùng bé $O((x-a)^{m+1})$).

Khai triển Taylor hàm 2 biến tại điểm (x_0, y_0):

$$\boldsymbol{f(x_0 + \Delta x, y_0 + \Delta y) = f(x_0, y_0) + df(x_0, y_0) + \frac{1}{2!}d^2 f(x_0, y_0) + \cdots + \frac{1}{n!}d^n f(x_0, y_0)}$$

$$\boldsymbol{+ \frac{1}{(n+1)!}d^{n+1} f(x_0 + \theta\Delta x, y_0 + \theta\Delta y), 0 < \theta < 1}$$

Khai triển Taylor giúp ta xấp xỉ được hàm về dạng chuỗi lũy thừa của đa thức – hàm rất tốt khi nó khả vi và khả tích mọi cấp.

4. Phương trình tích phân

Định nghĩa. Trong toán học phương trình tích phân là một phương trình trong đó một hàm số chưa biết xuất hiện dưới/trong dấu tích phân.

Giữa phương trình tích phân và phương trình vi phân có mối quan hệ khăng khít. Trên thực tế điều đó có nghĩa là cùng một hiện tượng trong tự nhiên người ta có thể dùng một trong hai công cụ này để mô tả một cách chính xác. Phương trình tích phân có thể được biểu diễn như sau:

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

$$y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} f\big(t, y(t)\big)dt$$

5. Các phương pháp số để giải phương trình vi phân thường

Thông thường người ta sử dụng các phương pháp số sau để giải phương trình vi phân, các phương pháp này sẽ cho ra kết quả dưới dạng hàm lưới:

- Phương pháp Euler
- Phương pháp Runge-Kutta
- Phương pháp ADAMs

…

Như đã nói ở phần mở đầu, trong phần tiếp theo của bài báo cáo này ta sẽ tìm hiểu về các công thức Runge-Kutta (R-K) từ ý tưởng, cách xây dựng công thức, đánh giá sai số, cho đến thiết lập thuật toán, chạy chương trình và đưa ra những đánh giá về phương pháp này qua các ví dụ.

B. CÁC CÔNG THỨC RUNGE-KUTTA

1. Ý tưởng phương pháp

Để giải số phương trình vi phân thường ta có thể sử dụng các phương pháp thuộc họ Euler như Euler hiện, Euler ẩn, công thức hình thang. Các phương pháp này có thể giải thô phương trình vi phân thường, ta vẫn cần tính toán với sai số nhỏ và ổn định hơn. Phương pháp Runge-Kutta được hai nhà toán học người Đức đề xuất ra để giải phương trình vi phân với sai số tốt và ổn định hơn. R-K sẽ tính tích phân trong phương trình tích phân qua s nấc trung gian mà vẫn đảm bảo việc tính thông qua các nấc trung gian có hiệu quả giống như khai triển Taylor hàm $y(x)$ đến bậc cao.

Ta đưa bài toán Cauchy về phương trình tích phân tương đương:

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

Giả sử ta đã biết giá trị của hàm y tại $x = x_n$ là y_n, ta cần tính $y(x_{n+1})$ thì ta có phương trình tích phân: $y(x_{n+1}) = y_n + \int_{x_n}^{x_{n+1}} f\big(t, y(t)\big)dt$. Ta đặt $\Delta y = \int_{x_n}^{x_{n+1}} f\big(t, y(t)\big)dt$

Ta sẽ tính Δy thông qua s nấc trung gian, có nghĩa là khi đó ta có: $y_{n+1} = y_n + r_1 k_1^{(n)} + r_2 k_2^{(n)} + \cdots + r_s k_s^{(n)}$ trong đó r_i là trọng số, các giá trị k_i được tính qua giá trị $f(x_i, y_i)$ (x_i, y_i được xác định qua các giá trị đã biết và tính được một cách trực tiếp gọi là phương pháp hiện) $k_i^{(n)} = hf\Big(x_n + \alpha_i h, y_n + \beta_{i1} k_1^{(n)} + \cdots + \beta_{ii-1} k_{i-1}^{(n)}\Big)$ với $\alpha_1 = 0$, $\alpha_i \in [0,1]$. Ta sẽ chọn các hệ số thích hợp để đạt được cấp sai số mà ta mong muốn.

2. Cách xây dựng các công thức R-K

a) Cách xây dựng tổng quát

Để giải bài toán Cauchy theo phương pháp R-K ta sẽ tìm nghiệm tại các điểm $x_1, x_2, \dots, x_n$ mà $a < x_1 < \cdots < x_n \le b$. Tại điểm x_k $(1 \le k \le n)$ ta tìm gần đúng giá trị nghiệm của phương trình, tức là tìm $y_k \approx y(x_k)$, y_k được tính thông qua 1 giá trị trước đó y_{k-1} nên phương pháp R-K là phương pháp 1 bước.

Xét bài toán Cauchy, trong đó hàm $f(x, y)$ thỏa mãn điều kiện Lipschitz theo biến y trên miền mở D:

$$\forall(x,y),(x,\overline{y}) \in D \qquad |f(x,y) - f(x,\overline{y})| \le M|y - \overline{y}|$$

Giả sử $(a, y_0) \in D$. Khi đó tồn tại duy nhất nghiệm $y^*(x)$ của bài toán và nghiệm có thể thác triển đến biên của D.

Bây giờ ta giả sử đã biết giá trị nghiệm $y(x)$ tại điểm x, làm thế nào để tìm nghiệm gần đúng $y(x+h)$ tại điểm $x+h$ thuộc miền đang xét. Đầu tiên, khai triển Taylor của hàm $y(x+h)$ tại x đến cấp m ta được:

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2!}y''(x) + \cdots + \frac{h^m}{m!}y^{(m)}(x) + \frac{h^{m+1}}{(m+1)!}y^{(m+1)}(c) \qquad (1)$$

trong đó $x \le c \le x+h$ còn $y^{(i)}(x)$ là đạo hàm cấp i của $y(x)$.

Ta đã biết $y(x)$ và $y'(x) = f(x,y)$ cho nên ta có thể tính được gần đúng giá trị của $y(x+h)$. Nghiệm càng chính xác nếu ta lấy m càng lớn, tuy nhiên ta phải tính giá trị của các hàm khác nhau tại một điểm, điều này nhìn chung có vẻ phức tạp.

Theo phương pháp R-K được đề xuất bởi hai nhà toán học người Đức C. Runge và M. W. Kutta thì ta chỉ phải tính hàm $f(x,y)$ tại một số điểm khác nhau. Trước tiên ta đặt $y(x+h) \approx z(h) = y(x) + \Delta y$, trong đó $\Delta y = r_1 k_1(h) + \cdots + r_s k_s(h)$ và

$$\begin{cases} k_i(h) = hf(\xi_i, v_i) & \xi_i = a + \alpha_i h;\ \alpha_1 = 0 \\ v_i = y_0 + \beta_{i1}k_1(h) + \cdots + \beta_{i,i-1}k_{i-1}(h) & \left(i = \overline{1,s}\right) \end{cases} \qquad (2)$$

Các hằng số $r_i, \alpha_i, \beta_{i,j}$ được chọn sao cho $\varphi(h) = y(x+h) - z(h)$ có sai số là ít nhất, tức là chọn sao cho $y(x+h)$ và $z(h)$ trùng nhau đến cấp cao nhất của h mà ta mong muốn.

Ta có $y' = f(x,y)$ (viết gọn là $y' = f$) thì khi đó ta tính các đạo hàm cấp cao của y được:

$$y'' = f'_x + f'_y . y'_x = f'_x + f'_y . f$$

$$y''' = f''_{x^2} + 2f.f''_{xy} + f^2.f''_{yy} + \left[f'_x + f.f'_y\right].f'_y$$

$$\begin{aligned} y'''' = {} & f'''_{x^3} + f'''_{x^2y}.f + 2\left(f'_x + f'_y.f\right)f''_{xy} + 2f\left[f'''_{x^2y} + f'''_{xy^2}.f\right] + 2f\left[f'_x + f'_y.f\right]f''_{yy} \\ & + f^2\left[f'''_{xy^2} + f'''_{y^3}.f\right] + \{f''_{x^2} + 2f.f''_{xy} + f^2.f''_{yy} + \left[f'_x + f.f'_y\right].f'_y\}.f'_y \\ & + \left[f'_x + f.f'_y\right]\left(f''_{xy} + f''_{yy}.f\right) \\ = {} & f'''_{x^3} + 3ff'''_{x^2y} + 3f'_xf''_{xy} + 4ff'_yf''_{xy} + 3f^2f'''_{xy^2} + 3ff'_xf''_{yy} + 4f^2f'_yf''_{yy} + f^3f'''_{y^3} \\ & + f'_yf''_{x^2} + f'_xf'_yf'_y + f.f'_yf'_yf'_y \end{aligned}$$

…

Tương tự ta tính được đạo hàm cấp $m+1$ của y thông qua đạo hàm cấp m của y. Tuy nhiên việc viết ra khá dài dòng nên ở đây ta chỉ dừng lại ở $m = 4$.

Thay vào (1) ta được $y(x+h) = y(x) + hf + \frac{h^2}{2!}\left[f'_x + f.f'_y\right] + \frac{h^3}{3!}\{f''_{x^2} + 2f.f''_{xy} + f^2.f''_{y^2} + \left[f'_x + f.f'_y\right].f'_y\} + \cdots$ (3)

Bên cạnh đó ta tính vi phân các cấp của hàm f, vì y là biến phụ thuộc x nên ta có vi phân các cấp của $f(x,y)$ là:

$$df = f'_x dx + f'_y dy,$$

$$d^2f = d(df) = \left(\frac{\partial}{\partial x}dx + \frac{\partial}{\partial y}dy\right)^2 f + f'_x dx^2 + f'_y dy^2$$

$$d^3f = d(d^2f) = \left(\frac{\partial}{\partial x}dx + \frac{\partial}{\partial y}dy\right)^3 f + \left(f'_x + 2f''_{x^2}\right)dx^3 + \left(f'_y + 2f''_{y^2}\right)dy^3$$
$$+3(dx^2dy + dxdy^2)f''_{xy}$$

......

Từ đó thay dx, dy bởi $\Delta x, \Delta y$ theo khai triển Taylor của hàm 2 biến, ta xấp xỉ được hàm $f(x+\Delta x, y+\Delta y) = f(x,y) + f'_x\Delta x + f'_y\Delta y\ +$ ..., từ (2) ta có:

$$k_1(h) = hf$$
$$k_2(h) = hf\left(x + \alpha_2 h, y + \beta_{21}k_1(h)\right) = h\left[f + \alpha_2 hf'_x + \beta_{21}k_1(h)f'_y\right] + \cdots$$
$$= h\left[f + \alpha_2 hf'_x + \beta_{21}hf.f'_y\right] + \cdots (4)$$

Ta có thể tiếp tục tính xấp xỉ $k_2(h)$ đến cấp cao hơn trong khai triển để h đạt sai số đến cấp mà ta mong muốn. Tương tự ta có thể tính $k_3(h), k_4(h)$, ... đến một vô cùng bé cấp nào đó.

Thay (4) vào Δy ta được $z(h) = y(x) + \Delta y$,

$$y(x) + \Delta y = y(x) + r_1k_1(h) + \cdots + r_sk_s(h) = r_1hf + r_2h\left[f + \alpha_2 hf'_x + \beta_{21}hf.f'_y\right] + \cdots$$

Đồng nhất hệ số với (3) để có hệ phương trình điều kiện của công thức R-K, dễ thấy rằng luôn có $r_1 + r_2 + \cdots + r_s = 1$.

b) Biểu diễn công thức R-K

**Butcher tableau:*

Các công thức R-K đều có dạng tổng quát như nhau và chỉ khác nhau ở các hệ số cho nên ta chỉ cần quan tâm đến hệ số là có thể xác định được công thức. Chính vì vậy mà người ta sử dụng một bảng giá trị để biểu diễn các hệ của một công thức R-K. Bảng giá trị có tên là hoạt cảnh Butcher (bảng Butcher), nó được đặt tên theo một nhà toán học người New Zealand – John.C.Butcher – người có đóng góp rất lớn trong nghiên cứu phương pháp R-K, bảng Bucher có dạng như sau:

$$\begin{array}{c|ccccc} \alpha_1 & \beta_{11} & \beta_{21} & \cdots & \beta_{1,s-1} & \beta_{1,s} \\ \alpha_2 & \beta_{21} & \beta_{21} & \cdots & \beta_{2,s-1} & \beta_{2,s} \\ \alpha_3 & \beta_{31} & \beta_{32} & \cdots & \beta_{3,s-1} & \beta_{3,s} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ \alpha_s & \beta_{s1} & \beta_{s2} & \cdots & \beta_{s,s-1} & \beta_{s,s} \\ \hline & r_1 & r_2 & \dots & r_{s-1} & r_s \end{array} \qquad \begin{array}{c|c} \alpha & \text{B} \\ \hline & \text{r} \end{array}$$

trong đó $\alpha_1 = 0$, α là ma trận cột kích thước s, r là ma trận hàng với kích thước s, B là ma trận vuông cấp s, đối với công thức R-K hiện thì bảng Butcher có B là ma trận tam giác dưới ($\beta_{i,j} = 0\ \forall i \le j$). Bởi vì mỗi hoạt cảnh Butcher đều biểu diễn cho một công thức R-K nên ta sẽ sử dụng hoạt cảnh Butcher để dễ dàng nhìn thấy các hệ số trong công thức mà ta đang nhắc đến mà không cần phải viết đầy đủ công thức. Vì ta chủ yếu nói về công thức R-K hiện nên B là ma trận tam giác dưới. Ví dụ như 2 hoạt cảnh thường dùng sau của R-K bậc 4.

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{3} & \frac{1}{3} & & & \\ \frac{2}{3} & -\frac{1}{3} & 1 & & \\ 1 & 1 & -1 & 1 & \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

3. Các công thức R-K thường dùng

a) Công thức R-K 1 nấc

Ta có $\Delta y = r_1 k_1(h); k_1(h) = hf(x_0, y_0)$, để công thức đạt sai số $O(h^2)$ thì theo như phần trước buộc phải có $r_1 = 1$, $\alpha_1 = 0$. Thật vậy, do

$\varphi(h) = h(1 - r_1)f\big(x, y(x)\big) + O(h^2)$ nên chỉ khi $r_1 = 1$ thì ta mới đạt sai số $O(h^2)$ và bảng Butcher của nó rất đơn giản là:

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

Đây là công thức R-K 1 nấc duy nhất, nó chính là công thức Euler. Sai số một bước là $O(h^2)$.

b) Công thức R-K 2 nấc

Ta có $y(x+h) \approx \mathrm{z(h)} = \mathrm{y(x)} + \mathrm{r}_1 k_1(h) + \mathrm{r}_2 k_2(h)$.

Ở đây cần tìm $r_1, r_2, \alpha_2, \beta_{21}$. Công thức cần đạt sai số ở cấp $O(h^3)$ nên theo (3) và (4) ở phần xây dựng tổng quát trước đó ta có:

$$\varphi(h) = h(1 - r_1 - r_2)f(x,y) + h^2\left(\frac{1}{2} - \alpha_2 r_2\right) + h^2\left(\frac{1}{2} - \beta_{21} r_2\right) f_y'.f + O(h^3)$$

$$\varphi(h) = O(h^3) \text{ chỉ khi } \begin{cases} r_1 + r_2 = 1 \\ \alpha_2 r_2 = \dfrac{1}{2} \\ \beta_{21} r_2 = \dfrac{1}{2} \end{cases}$$

Đây là hệ gồm 3 phương trình 4 ẩn, do đó nó là hệ vô định. Vì ta muốn có công thức dễ tính toán nên ta chọn vài trường hợp mà các giá trị của $\alpha_2, r_1, r_2, \beta_{21}$ đơn giản mà thỏa mãn hệ phương trình và đều đạt sai số cấp $O(h^3)$.

Ta có thể chọn như sau:

+ Chọn $r_1 = 0, r_2 = 1 \Rightarrow \alpha_2 = \beta_{21} = \frac{1}{2}$

+ Chọn $r_1 = r_2 = \frac{1}{2} \Rightarrow \alpha_2 = \beta_{21} = 1$

Chọn $\alpha_2 = a$ là tham số, ta được các công thức R-K2 khác nhau theo a

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array} \qquad \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|cc} 0 & & \\ a & a & \\ \hline & 1 - \frac{1}{2a} & \frac{1}{2a} \end{array}$$

Công thức được biểu diễn ở bảng giữa còn được gọi là công thức Heun bậc 2.

c) Công thức R-K 3 nấc

Với 3 nấc ta cũng làm tương tự như trường hợp 2 nấc.

Ta có $y(x+h) \approx z(h) = y(x) + r_1 k_{1(h)} + r_2 k_2(h) + r_3 k_3(h)$. Ở đây ta cần tìm 8 giá trị $r_1, r_2, r_3, \alpha_2, \alpha_3, \beta_{21}, \beta_{31}, \beta_{32}$. Ta sẽ làm tương tự như phần xây dựng tổng quát nhưng cụ thể hơn, ta sẽ dừng ở cấp $O(h^4)$:

Ta có: $y(x+h) = y(x) + hf + \frac{h^2}{2}\left[f'_x + f'_y . f\right]$

$$+\frac{h^3}{6}\left[f''_{x^2} + 2f.f''_{xy} + f^2.f''_{yy} + \left[f'_x + f.f'_y\right].f'_y\right] + O(h^4)$$

Và $k_1(h) = hf$

$$k_2(h) = hf + h^2\left(\alpha_2 f'_x + \beta_{21} f f'_y\right) + h^3\left(\tfrac{1}{2}\alpha_2^2 f''_{xx} + \alpha_2\beta_{31} f f''_{xy} + \tfrac{1}{2}\beta_{21}^2 f^2 f''_{yy}\right) + \cdots$$

$$k_3(h) = h\Big[f + \alpha_3 h f'_x + (\beta_{31}k_1 + \beta_{32}k_2) f'_y + \frac{h^2}{2}\alpha_3^2 f''_{xx} + \alpha_3 h(\beta_{31}k_1 + \beta_{32}k_2) f''_{xy} + (\beta_{31}k_1 + \beta_{32}k_2)^2 f''_{yy} + \cdots\Big]$$

Khi đó ta có hệ phương trình sau:
$$\begin{cases} r_1 + r_2 + r_3 = 1 \\ r_2\alpha_2 + r_3\alpha_3 = \dfrac{1}{2} \\ r_2\beta_{21} + r_3(\beta_{31} + \beta_{32}) = \dfrac{1}{2} \\ \dfrac{1}{2}r_2\alpha_2^2 + \dfrac{1}{2}r_3\alpha_3^2 = \dfrac{1}{6} \\ \dfrac{1}{2}r_2\alpha_2^2 + \dfrac{1}{2}r_3\alpha_3(\beta_{31} + \beta_{32}) = \dfrac{1}{6} \\ \dfrac{1}{2}r_2\alpha_2^2 + \dfrac{1}{2}r_3(\beta_{31}^2 + \beta_{31}\beta_{32} + \beta_{32}^2) = \dfrac{1}{6} \\ r_3\beta_{32}\alpha_2 = \dfrac{1}{6} \\ r_3\beta_{21}\beta_{32} = \dfrac{1}{6} \end{cases}$$

Đến đây ta cũng có thể chọn một vài bộ giá trị thỏa mãn hệ phương trình:

- Chọn $r_1 = r_3 = \frac{1}{6}, r_2 = \frac{2}{3}, \alpha_2 = \frac{1}{2}, \alpha_3 = 1, \beta_{21} = \frac{1}{2}, \beta_{31} = -1, \beta_{32} = 2$ ta có công thức R-K 3 nấc thường dùng, công thức Simpson.

- Chọn $r_1 = \frac{1}{4}, r_2 = 0, r_3 = \frac{3}{4}, \alpha_2 = \beta_{21} = \frac{1}{3}, \alpha_3 = \beta_{32} = \frac{2}{3}, \beta_{31} = 0$ ta có một công thức khác thường được gọi là công thức Heun bậc 3.

Bảng Butcher của 2 trường hợp này lần lượt là:

0	(Simpson)		
$\frac{1}{2}$	$\frac{1}{2}$		
1	-1	2	
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

0	(Heun)		
$\frac{1}{3}$	$\frac{1}{3}$		
$\frac{2}{3}$	0	$\frac{2}{3}$	
	$\frac{1}{4}$	0	$\frac{3}{4}$

d) Công thức R-K 4 nấc

Với 4 nấc ta lại tiếp tục xây dựng tương tự các trường hợp trước đó. Ta có:
$\varphi_4(h) = y(x_0 + h) - y(x_0) - \sum_{i=1}^{4} r_i\, k_i(h)$
Dễ thấy $\varphi_4(0) = 0; \varphi_4'(0) = y_0' - \sum_{i=1}^{4} r_i\, k_i'(0) = (1 - \sum_{i=1}^{4} p_{4i}) f_0 = 0$
Từ đây suy ra $\sum_{i=1}^{4} r_i = 1$. Ngoài ra ta còn đòi hỏi $\varphi_4''(0) = \varphi_4'''(0) = \varphi_4^{(4)}(0) = 0$. Việc tính toán các đạo hàm và vi phân cấp cao để đưa ra điều kiện cho công thức bậc 4 là khá dài nên người ta đã sử dụng đến lý thuyết đồ thị để cho việc xác định điều kiện được dễ dàng hơn.

Nói chung, ta sẽ có một hệ phương trình với 13 ẩn α_2, α_3, α_4, β_{21}, β_{31}, β_{32}, $\beta_{41}, \beta_{42}, \beta_{43}$, r_1, r_2, r_3, r_4. Từ đây tìm được một họ các công thức R-K4, người ta hay dùng 2 công thức có bảng Butcher như sau:

0				
$\frac{1}{3}$	$\frac{1}{3}$			
$\frac{2}{3}$	$-\frac{1}{3}$	1		
1	1	-1	1	
	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Công thức bên phải là công thức thông dụng nhất. Công thức này đạt sai số một bước là $O(h^5)$ và thường được sử dụng vì độ chính xác cao và dễ tính toán do trong bảng Butcher có nhiều số 0. Ngoài ra người ta cũng dùng thêm công thức R-K4 có tên thường gọi là luật 3/8 với bảng Butcher bên trái.

Ta cũng có thể xây dựng công thức R-K bậc cao hơn từ các công thức R-K bậc thấp hơn, ví dụ ta xây dựng công thức R-K bậc 4 cổ điển như sau:

Từ công thức RK3 (Simpson) đã nói ở phần trước ta có:

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

Lại có $y\left(x+\frac{h}{2}\right) \approx y(x) + \frac{h}{2}f$ (công thức RK1)

$$y\left(x+\frac{h}{2}\right) \approx y(x) + \frac{h}{2}f\left(x+\frac{h}{2}, y\left(x+\frac{h}{2}\right)\right) \text{ (Euler dạng ẩn kết hợp R-K1)}$$

$$y(x+h) \approx y(x) + hf\left(x+\frac{h}{2}, y\left(x+\frac{h}{2}\right)\right) \text{ (công thức trung điểm)}$$

Từ đây thay vào nhau và ta có

$$y(x+h) \approx y(x) + \frac{h}{6}\left[f + 2f\left(x+\frac{h}{2}, y+\frac{h}{2}f\right) + 2f\left(x+\frac{h}{2}, y+hf\left(x+\frac{h}{2}, y+\frac{h}{2}f\right)\right) + f\left(x+h, y+hf\left(x+\frac{h}{2}, y+hf\left(x+\frac{h}{2}, y+\frac{h}{2}f\right)\right)\right)\right].$$

Đây chính là công thức R-K4 cổ điển.

e) Các công thức R-K nhiều hơn 4 nấc

Sai số của các công thức 4 nấc cũng đã khá tốt cho nên công thức R-K4 thường được sử dụng nhiều.

Với nhiều nấc hơn ta thu được các công thức R-K đạt độ chính xác cao hơn song việc tính toán là quá phức tạp. Việc tìm ra các điều kiện của hệ số trong công thức bậc cao cũng khó khăn hơn so với số nấc nhỏ hơn 5, để có được điều kiện của hệ số bậc cao (hệ phương trình liên hệ giữa các hệ số) ta có thể cần đến vài chục hay thậm chí vài trăm, vài nghìn điều kiện.

Khi xây dựng công thức R-K 5 nấc người ta nhận thấy rằng không thể đạt sai số cục bộ cấp $O(h^6)$ nếu chỉ tính thông qua 5 nấc trung gian. Ta có nhận xét sau:

***<u>Nhận xét</u>**:* Để có thể đạt được sai số cấp q thì số lượng nấc trung gian s của các công thức R-K nhiều hơn 4 nấc phải có $s > q$ và người ta thống kê rằng:

q	1	2	3	4	5	6	7	8	9	10
min s	1	2	3	4	6	7	9	11	13	≤17

Qua bảng này thì ta có thể thấy rằng nếu muốn có công thức sai số bậc 5 ta cần tính thông qua ít nhất 6 nấc trung gian. Muốn có sai số bậc 4 thì cần ít nhất 4 nấc trung gian, có nghĩa là ta có thể sử dụng nhiều hơn 4 nấc trung gian để có sai số bậc 4.

***Sau đây là một số bảng Butcher của công thức R-K s nấc với s lớn hơn 4:**

+ R-K 6 nấc:

$$
\begin{array}{c|cccccc}
0 & & & & & & \\
\frac{1}{4} & \frac{1}{2} & & & & & \\
\frac{1}{4} & \frac{3}{16} & \frac{1}{16} & & & & \\
\frac{1}{2} & \frac{-1}{4} & \frac{-1}{4} & 1 & & & \\
\frac{3}{4} & \frac{3}{16} & 0 & 0 & \frac{9}{16} & & \\
1 & \frac{-2}{7} & \frac{1}{7} & \frac{12}{7} & \frac{-12}{7} & \frac{8}{7} & \\
\hline
 & \frac{7}{90} & 0 & \frac{16}{45} & \frac{2}{15} & \frac{16}{45} & \frac{7}{90}
\end{array}
\qquad
\begin{array}{c|cccccc}
0 & & & & & & \\
\frac{1}{4} & \frac{1}{4} & & & & & \\
\frac{1}{4} & \frac{1}{8} & \frac{1}{8} & & & & \\
\frac{1}{2} & 0 & 0 & \frac{1}{2} & & & \\
\frac{3}{4} & \frac{3}{16} & \frac{-3}{8} & \frac{3}{8} & \frac{9}{16} & & \\
1 & \frac{-3}{7} & \frac{8}{7} & \frac{6}{7} & \frac{-12}{7} & \frac{8}{7} & \\
\hline
 & \frac{7}{90} & 0 & \frac{16}{45} & \frac{2}{15} & \frac{16}{45} & \frac{7}{90}
\end{array}
$$

+ R-K 9 nấc:

$$
\begin{array}{c|ccccccccc}
0 & & & & & & & & & \\
\frac{1}{6} & \frac{1}{6} & & & & & & & & \\
\frac{1}{3} & 0 & \frac{1}{3} & & & & & & & \\
\frac{1}{2} & \frac{1}{8} & 0 & \frac{3}{8} & & & & & & \\
\frac{2}{11} & \frac{148}{1331} & 0 & \frac{150}{1331} & \frac{-56}{1331} & & & & & \\
\frac{2}{3} & \frac{-404}{243} & 0 & \frac{-170}{27} & \frac{4024}{1701} & \frac{10648}{1701} & & & & \\
\frac{6}{7} & \frac{2466}{2401} & 0 & \frac{1242}{343} & \frac{-19176}{16807} & \frac{-51909}{16807} & \frac{1053}{2401} & & & \\
0 & \frac{5}{154} & 0 & 0 & \frac{96}{539} & \frac{-1815}{20384} & \frac{-405}{2464} & \frac{49}{1144} & & \\
1 & \frac{-113}{32} & 0 & \frac{-195}{22} & \frac{32}{7} & \frac{29403}{3584} & \frac{-729}{512} & \frac{1029}{1408} & \frac{21}{16} & \\
\hline
 & 0 & 0 & 0 & \frac{32}{105} & \frac{1771561}{6289920} & \frac{243}{2560} & \frac{16807}{74880} & \frac{77}{1440} & \frac{11}{270}
\end{array}
$$

\+ R-K 11 nấc:

$$
\begin{array}{c|ccccccccccc}
0 & & & & & & & & & & & \\
\frac{1}{2} & \frac{1}{2} & & & & & & & & & & \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & & & & & & & & & \\
\frac{7+\sqrt{21}}{14} & \frac{1}{7} & \frac{-7-3\sqrt{21}}{98} & \frac{21+5\sqrt{21}}{49} & & & & & & & & \\
\frac{7+\sqrt{21}}{14} & \frac{11+\sqrt{21}}{84} & 0 & \frac{18+4\sqrt{21}}{63} & \frac{21-\sqrt{21}}{252} & & & & & & & \\
\frac{1}{2} & \frac{5+\sqrt{21}}{48} & 0 & \frac{9+\sqrt{21}}{36} & \frac{-231+14\sqrt{21}}{360} & \frac{63-7\sqrt{21}}{80} & & & & & & \\
\frac{7-\sqrt{21}}{14} & \frac{10-\sqrt{21}}{42} & 0 & \frac{-432+92\sqrt{21}}{315} & \frac{633-145\sqrt{21}}{90} & \frac{-504+115\sqrt{21}}{70} & \frac{63-13\sqrt{21}}{35} & & & & & \\
\frac{7-\sqrt{21}}{14} & \frac{1}{14} & 0 & 0 & 0 & \frac{14-3\sqrt{21}}{126} & \frac{13-3\sqrt{21}}{63} & \frac{1}{9} & & & & \\
\frac{1}{2} & \frac{1}{32} & 0 & 0 & 0 & \frac{91-21\sqrt{21}}{576} & \frac{11}{72} & \frac{-385-75\sqrt{21}}{1152} & \frac{63+13\sqrt{21}}{128} & & & \\
\frac{7+\sqrt{21}}{14} & \frac{1}{14} & 0 & 0 & 0 & \frac{1}{9} & \frac{-733-147\sqrt{21}}{2205} & \frac{515+111\sqrt{21}}{504} & \frac{-51-11\sqrt{21}}{56} & \frac{132+28\sqrt{21}}{245} & & \\
1 & 0 & 0 & 0 & 0 & \frac{-42+7\sqrt{21}}{18} & \frac{-18+28\sqrt{21}}{45} & \frac{-273-53\sqrt{21}}{72} & \frac{301+53\sqrt{21}}{72} & \frac{28-28\sqrt{21}}{45} & \frac{49-7\sqrt{21}}{18} & \\
\hline
 & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20}
\end{array}
$$

Công thức Cooper-Verner (R-K có cấp chính xác 8)

4. Dùng các công thức R-K trong giải hệ phương trình vi phân

Như đã nói ở phần bài toán Cauchy, bằng một phép đổi biến hàm, ta có thể đưa một ODE cấp k trong không gian 1 chiều thành ODE cấp 1 trong không gian k chiều. Ta có thể sử dụng các công thức R-K trong không gian k chiều để giải phương trình và hệ phương trình vi phân thường. Các chứng minh cũng hoàn toàn tương tự như trong trường hợp 1 chiều và cũng có các tính chất, đặc điểm tương tự.

Với $x = [x_0, x_1, x_2, \ldots x_{k-1}]^{\mathrm{T}} \in \mathbb{R}^k$ là hàm vector theo biến t trên đoạn $[a, b]$, ta có hệ phương trình được biểu diễn đơn giản là $x' = f(t, x)$, từ đây về mặt kí hiệu là giống nhau, áp dụng công thức R-K tương tự như trường hợp 1 chiều để tính toán.

Ví dụ, xét bài toán Cauchy với một ODE cấp 2 như sau:

$$\begin{cases} y'' = t + y + y' \\ y'(0) = 1, y(0) = 0 \end{cases}$$

Từ đây nếu đặt $x_0 = y, x_1 = y', x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$ thì ta có:

$$x' = \begin{bmatrix} x_1 \\ t + x_0 + x_1 \end{bmatrix} = \begin{bmatrix} f(t, x_0, x_1) \\ g(t, x_0, x_1) \end{bmatrix} = F(t, x) \text{ và } x(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

ta cần xác định x_0 và x_1. Giả sử giá trị của 2 hàm này tại t_i lần lượt là x_{0i}, x_{1i} đã biết, cần tính $x_{0,i+1}, x_{1,i+1}$, ta sẽ áp dụng công thức R-K4 cổ điển như trong trường hợp ODE cấp 1:

$$k_{0,1} = hf(t_i, x_{0i}, x_{1i}) \qquad k_{1,1} = hg(t_i, x_{0i}, x_{1i})$$

$$k_{0,2} = hf\left(t_i + \frac{h}{2}, x_{0i} + \frac{k_{0,1}}{2}, x_{1i} + \frac{k_{1,1}}{2}\right) \qquad k_{1,2} = hg\left(t_i + \frac{h}{2}, x_{0i} + \frac{k_{0,1}}{2}, x_{1i} + \frac{k_{1,1}}{2}\right)$$

$$k_{0,3} = hf\left(t_i + \frac{h}{2}, x_{0i} + \frac{k_{0,2}}{2}, x_{1i} + \frac{k_{1,2}}{2}\right) \qquad k_{1,3} = hg\left(t_i + \frac{h}{2}, x_{0i} + \frac{k_{0,2}}{2}, x_{1i} + \frac{k_{1,2}}{2}\right)$$

$$k_{0,4} = hf(t_i + \mathrm{h}, x_{0i} + k_{0,3}, x_{1i} + \mathrm{k}_{1,3}) \qquad k_{1,4} = hg(t_i + h, x_{0i} + k_{0,3}, x_{1i} + k_{1,3})$$

$$x_{0,i+1} = x_{0,i} + \frac{k_{0,1} + 2k_{0,2} + 2k_{0,3} + k_{0,4}}{6} \qquad x_{1,i+1} = x_{1,i} + \frac{k_{1,1} + 2k_{1,2} + 2k_{1,3} + k_{1,4}}{6}$$

Với $0 \le t \le 0.9, h = 0.1$ thu được kết quả như sau:

t	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
y'	0	0.1055	0.2243	0.36	0.5169	0.7	0.915	1.1684	1.4683	1.8235
y	1	1.1157	1.2658	1.4556	1.6912	1.9799	2.3303	2.7526	3.2588	3.8631

5. Sai số của công thức Runge-Kutta

a) Sai số địa phương (Local Truncation Error)

Giả sử giá trị của hàm tại $x = x_n$ là y_n là kết quả đúng thì khi tính toán y_{n+1} bằng công thức R-K sẽ có sai số cắt cụt gọi là sai số địa phương (sai số cục bộ).

Gọi công thức R-K là có bậc p (R-K cấp p) nếu sai số địa phương là $O(h^{p+1})$. Xuất phát từ chính ý tưởng của phương pháp, đó là chọn các hệ số để trùng khớp với khai triển Taylor với cấp theo mong muốn.

b) Sai số toàn cục (Global Truncation Error)

Sai số toàn cục là sai số giữa kết quả đúng và kết quả số, sai số toàn cục của công thức R-K bậc q là $O(h^q)$. Ta có thể chứng minh được với công thức R-K cho ODE thì sai số toàn cục luôn lớn hơn sai số cục bộ 1 bậc.

Sau đây ta sẽ chứng minh sai số toàn cục cho R-K1 và một công thức RK2.

Bổ đề: Cho $\{x_n\}$ là dãy số thực thỏa mãn với mọi n rằng:

$|x_{n+1}| \le (1+\delta)|x_n| + \beta$ với hằng số $\delta > 0, \beta \ge 0$ thì ta có:

$$|x_n| \le e^{n\delta}|x_0| + \frac{e^{n\delta}-1}{\delta}\beta \quad \forall n$$

Chứng minh: Ta có $|x_{n+1}| \le (1+\delta)|x_n| + \beta, |x_n| \le (1+\delta)|x_{n-1}| + \beta, |x_{n-1}| \le (1+\delta)|x_{n-2}| + \beta, \ldots$ nên:

$|x_{n+1}| \le (1+\delta)[(1+\delta)|x_{n-1}| + \beta] + \beta = (1+\delta)^2|x_{n-1}| + (1+\delta)\beta + \beta$

$\le (1+\delta)^2[(1+\delta)|x_{n-2}| + \beta] + (1+\delta)\beta + \beta$

$= (1+\delta)^3|x_{n-2}| + (1+\delta)^2\beta + (1+\delta)\beta + \beta$

$\le \cdots \le (1+\delta)^{n+1}|x_0| + \beta\sum_{i=0}^{n}(1+\delta)^i = (1+\delta)^{n+1}|x_0| + \beta\frac{(1+\delta)^{n+1}-1}{1+\delta-1}$

$= (1+\delta)^{n+1}|x_0| + \beta\frac{(1+\delta)^{n+1}-1}{\delta} \qquad (*)$

Mặt khác $ln(1+\delta) \le \delta \Rightarrow (n+1)\ln(1+\delta) \le (n+1)\delta \Rightarrow (1+\delta)^{n+1} \le e^{(n+1)\delta}$

Nên từ (*) ta có $|x_{n+1}| \le e^{(n+1)\delta}|x_0| + \frac{e^{(n+1)\delta}-1}{\delta}\beta$. Bổ đề được chứng minh.

Gọi y_n là nghiệm giải bởi RK tại x_n, $y(x_n)$ là nghiệm chính xác tại x_n

Với RK1: $y_{n+1} = y_n + hf(x_n, y_n),\ \ y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(\xi)$

$$e_{n+1} = y_{n+1} - y(x_{n+1}) = y_n - y(x_n) + hf(x_n, y_n) - hf\big(x_n, y(x_n)\big) - \frac{y''(\xi)}{2}h^2$$

Lipschitz $\Rightarrow |e_{n+1}| \le |e_n| + hL|y_n - y(x_n)| + Mh^2 = (1+Lh)|e_n| + Mh^2$

Áp dụng bổ để với $\delta = Lh,\ \beta = Mh^2 \Rightarrow |e_n| \le e^{nLh}|e_0| + Mh^2\frac{e^{nLh}-1}{Lh}$

Sai số đầu vào $e_0 = 0 \Rightarrow$ RK1 có sai số toàn cục O(h)

Với RK2, xét công thức :

$$y_{n+1} = y_n + \frac{hf(x_n, y_n)}{2} + \frac{hf\big(x_n + h, y_n + hf(x_n, y_n)\big)}{2}$$

$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{3!}y'''(\xi)$

$e_{n+1} = y_{n+1} - y(x_{n+1})$

$= y_n - y(x_n) + \frac{hf(x_n, y_n)}{2} + \frac{hf\big(x_n + h, y_n + hf(x_n, y_n)\big)}{2} - hf\big(x_n, y(x_n)\big)$

$- \frac{\big[f'_x\big(x_n, y(x_n)\big) + f'_y\big(x_n, y(x_n)\big)f\big(x_n, y(x_n)\big)\big]h^2}{2} - \frac{h^3}{3}y'''(\xi)$

$= e_n + \frac{hf(x_n,y_n)}{2} + \frac{hf(x_n,y_n)}{2} + \frac{h^2}{2}\big[f'_x(x_n, y_n) + f'_y(x_n, y_n)f(x_n, y_n) + O(h)\big] - hf\big(x_n, y(x_n)\big)$

$$-\frac{[f'_x(x_n,y(x_n)) + f'_y(x_n,y(x_n))f(x_n,y(x_n))]h^2}{2} - \frac{h^3}{3}y'''(\xi)$$

$$= e_n + hf(x_n,y_n) - hf(x_n,y(x_n)) + \frac{h^2}{2}[f'_x(x_n,y_n) - f'_x(x_n,y(x_n))]$$

$$+\frac{h^2}{2}[f'_y(x_n,y_n)f(x_n,y_n) - f'_y(x_n,y(x_n))f(x_n,y(x_n)) + O(h)] - \frac{h^3}{3}y'''(\xi)$$

Đặt $g(x,y) = f'_x(x,y), k(x,y) = f(x,y)f'_y(x,y)$

Với $|g'_y| \le M_1$, $|k'_y| \le M_2$, $M_1 + M_2 = M^2$

$$\Rightarrow |e_{n+1}| \le |e_n| + Lh|e_n| + \frac{M^2h^2}{2}|e_n| + Ch^3 = \left(1 + Lh + \frac{(Mh)^2}{2}\right)|e_n| + \mathrm{C}h^3$$

Áp dụng bổ đề với $\delta = Lh + \frac{(Mh)^2}{2}$, $\beta = Ch^3$

$$\Rightarrow |e_n| \le e^{n\left(Lh+\frac{(Mh)^2}{2}\right)}|e_0| + \mathrm{C}h^3\,\frac{e^{n\left(Lh+\frac{(Mh)^2}{2}\right)}-1}{Lh+\frac{Mh^2}{2}} \text{ với } e_0 = 0 \Rightarrow |e_n| \le Kh^2$$

Công thức này có sai số toàn cục $O(h^2)$

6. Miền ổn định của công thức R-K

Xét phương trình thử $y' = \lambda y, Re(\lambda) < 0$, đặt $z = h\lambda$. Miền giá trị của z mà công thức có thể hội tụ ($y_n \to 0$) thì gọi là ổn định. Hàm ổn định là $R(z)$, miền ổn định được định nghĩa là tập $\{z \in \mathbb{C}: |R(z)| \le 1\}$.

- Với $p = 1$ ta có công thức Euler, với phương trình thử $y' = \lambda y$, có: $y_{n+1} = y_n + hf(x_n,y_n) = (1 + h\lambda)y_n = (1 + h\lambda)^{n+1}y_0$

$\Rightarrow \{z \in \mathbb{C}: |1 + z| < 1\}$ là miền ổn định của công thức.

- Với $p = 2$, chọn công thức Heun cấp 2, có:

$$y_{n+1} = y_n + \frac{h}{2}f(x_n,y_n) + \frac{h}{2}f(x_n + \mathrm{h}, y_n + hf(x_n,y_n))$$

$$= y_n + \frac{h}{2}\lambda y_n + \frac{h}{2}\lambda(y_n + h\lambda y_n) = y_n\left(1 + \mathrm{z} + \frac{\mathrm{z}^2}{2}\right) = \left(1 + \mathrm{z} + \frac{\mathrm{z}^2}{2}\right)^{\mathrm{n}+1} \mathrm{y}_0$$

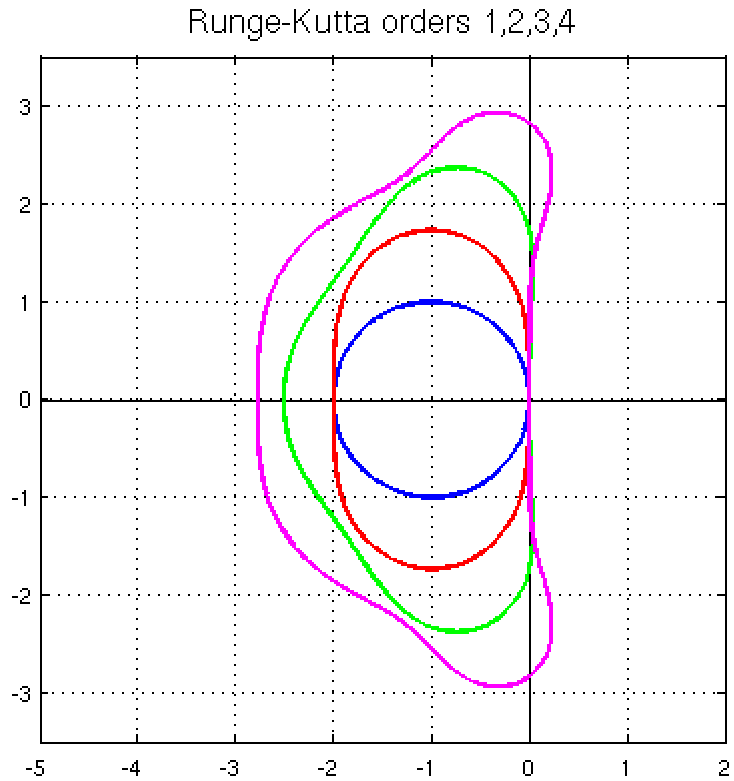
$\Rightarrow \{z \in \mathbb{C}: \left|1 + \mathrm{z} + \frac{z^2}{2}\right| < 1\}$ là miền ổn định của công thức này.

- Với $p = 3, p = 4$ xét 2 công thức, từ đây có một số hàm ổn định:

$$R(z) = \begin{cases} 1 + z, & p = 1 \\ 1 + z + \frac{1}{2}z^2, & p = 2 \\ 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3, & p = 3 \\ 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4, & p = 4 \end{cases}$$

Đối với công thức R-K hiện bậc cao hơn 2 thì miền ổn định rộng hơn miền ổn định của công thức Euler hiện, miền ổn định này là khá tốt cho việc ta chọn *h* khi thực hiện phương pháp. Miền ổn định vẫn còn chưa đủ tốt nếu ta có yêu cầu cao hơn. Từ miền ổn định này, khi giải phương trình vi phân cần chọn h hợp lý để công thức ổn định và thỏa mãn sai số.

Dưới đây là đồ thị về miền ổn định của công thức R-K trong 4 trường hợp kể trên:



Miền ổn định của một số công thức R-K

7. Đánh giá phương pháp

Phương pháp Runge-Kutta có những ưu và nhược điểm như sau:

+ Ưu điểm:

- Có đủ độ chính xác và ổn định cho nhiều bài toán.
- Thuật toán đơn giản, không phức tạp, dễ thực hiện.
- Không phải tính giá trị của nhiều hàm khác nhau tại 1 điểm mà chỉ cần tính giá trị của 1 hàm tại một vài điểm.

+ Nhược điểm:

- Với bước cố định, các giá trị gần điểm bắt đầu có độ chính xác cao, càng ra xa càng kém.
- Chi phí tính toán lớn.
- Cần phải chọn bước h hợp lý để nghiệm hội tụ.

Ta sẽ thử nghiệm trong phần ví dụ của thuật toán.

8. Phương pháp Runge-Kutta bước thích ứng

Đặt ra một câu hỏi là nếu muốn giải với sai số ε cho trước thì phải làm như thế nào? Nhận thấy rằng nếu sử dụng bước h nhỏ thì sai số sẽ nhỏ hơn, và nếu dùng công thức bậc cao hơn thì sai số cũng nhỏ hơn. Nếu chỉ chọn ngẫu nhiên bước h ban đầu và chia lưới đều thì sẽ không đảm bảo được sai số. Vì thế nên có phương pháp thay đổi h trong lúc tính gọi là Runge-Kutta bước thích ứng. Có 2 cách thường được sử dụng để thay đổi h đó là dùng 2 công thức R-K có bậc sai số là p và $p+1$ hoặc dùng một công thức cấp p nhưng tính với bước h và $\frac{h}{2}$.

- Với cách thứ hai gọi là phương pháp chia đôi bước tích phân, ở mỗi lần ta sẽ tính theo 2 bước là h và $\frac{h}{2}$, gọi y_h và $y_{\frac{h}{2}}$ là giá trị tính bằng bước $h, \frac{h}{2}$.

Giả sử dùng công thức RK4, khi đó $y_h = y_n + ch^5 + O(h^6), y_{\frac{h}{2}} = y_n + 2c\left(\frac{h}{2}\right)^5 + O(h^6)$, tính $\Delta = \left|y_h - y_{\frac{h}{2}}\right| \sim \frac{15}{16}h^5$ dùng điều này để đánh giá, sai số nếu dùng bước h là vượt quá mức chấp nhận được $(\Delta > \varepsilon)$ thì ta sẽ chia đôi bước h và lặp lại như vậy, nếu tỉ lệ nằm trong mức chấp nhận thì tiếp tục dùng h, lấy $y_{n+1} = y_{\frac{h}{2}} + \frac{\Delta}{15}$.

- Với cách thứ nhất gọi là phương pháp Runge-Kutta-Fehlberg (RKF), sai số của công thức cấp $p+1$ sẽ tốt hơn công thức cấp p và coi như rất bé, tính toán giá trị chênh lệch giữa 2 sai số cấp p và $p+1$ tại mỗi thời điểm và thay đổi h để phù hợp với tỉ lệ sai số. Ví dụ với 2 công thức có sai số cục bộ là $O(h^5)$ và $O(h^6)$ khi lấy hiệu 2 giá trị y_{n+1} tính qua y_n theo 2 công thức này ta được $\delta = \left|y_{n+1}^{(5)} - y_{n+1}^{(6)}\right| \sim h^5$. Từ đây ta đưa ra đánh giá và điều chỉnh bước h như sau $h_{new} = h_{old}\left(\frac{\varepsilon}{\delta}\right)^{\frac{1}{5}}$, theo như công thức này, nếu $\delta > \varepsilon$ tức là chưa đạt sai số mong muốn thì h sẽ được giảm đi, tương tự với $\delta \leq \varepsilon$.

Để tính toán ít phức tạp, chọn 2 công thức đạt sai số cấp p và $p+1$ sao cho hệ số α và β trong bảng Butcher là giống nhau, chỉ khác nhau hệ số r. Bảng Butcher của phương pháp RKF này sẽ có thêm một hàng r^* dưới hàng r, nếu dùng hàng r^* thì sai số là cấp p, nếu dùng hàng r thì sai số là cấp $p+1$.

$$
\begin{array}{c|cccccc}
0 & & & & & & \\
\frac{1}{4} & \frac{1}{4} & & & \text{(Fehlberg)} & & \\
\frac{3}{8} & \frac{3}{32} & \frac{9}{32} & & & & \\
\frac{12}{13} & \frac{1932}{2197} & -\frac{7200}{2197} & \frac{7296}{2197} & & & \\
1 & \frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} & & \\
\frac{1}{2} & -\frac{8}{27} & 2 & -\frac{3544}{2565} & \frac{1859}{4104} & -\frac{11}{40} & \\
\hline
 & \frac{16}{135} & 0 & \frac{6656}{12825} & \frac{28561}{56430} & -\frac{9}{50} & \frac{2}{55} \\
 & \frac{25}{216} & 0 & \frac{148}{2565} & \frac{2197}{4104} & -\frac{1}{5} & 0
\end{array}
$$

Bảng Butcher của phương pháp Fehlberg

C. THUẬT TOÁN, VÍ DỤ

1. Thuật toán

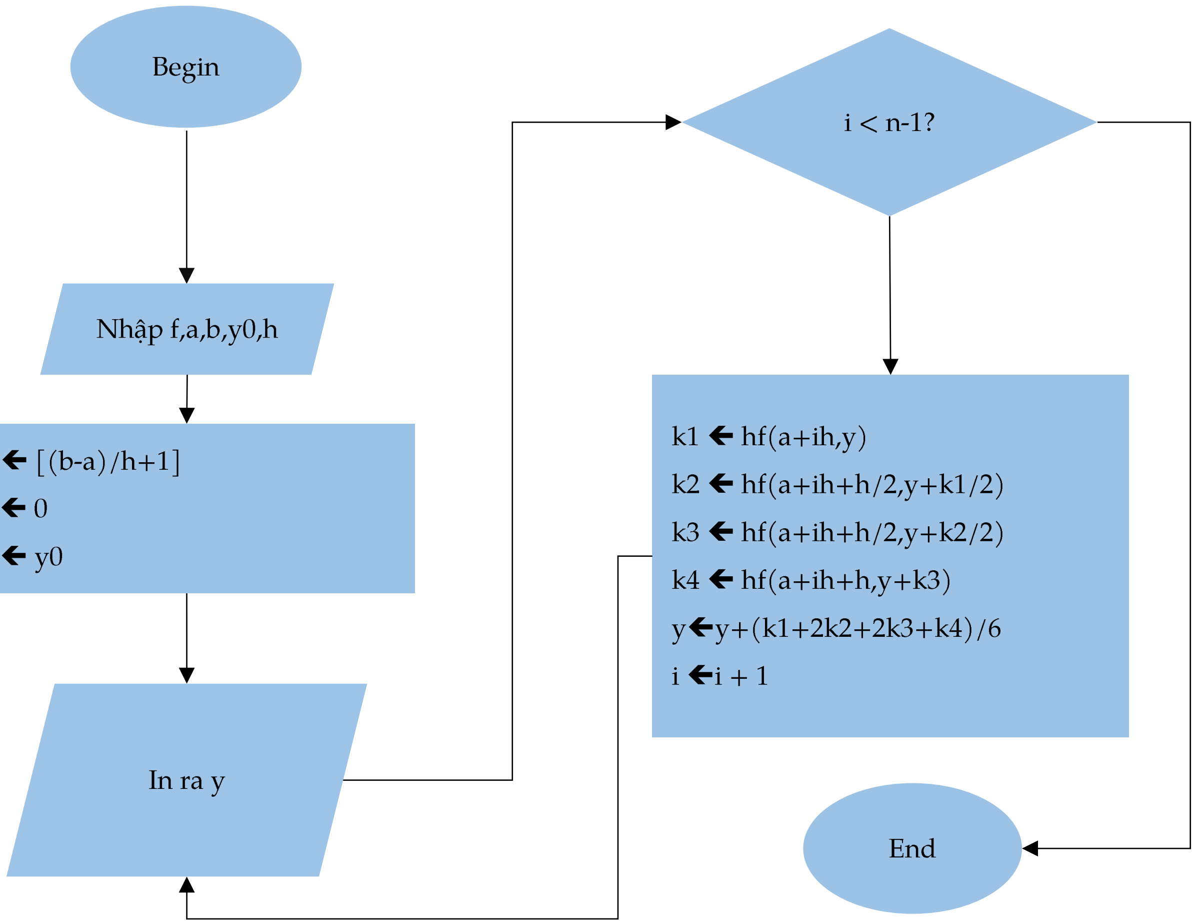
Các công thức R-K có dạng tương tự như nhau và cũng dễ triển khai bằng các ngôn ngữ lập trình khác nhau nên ở đây ta sẽ biểu diễn thuật toán của một công thức R-K cụ thể đó là R-K4 thông dụng nhất, từ đó các công thức khác cũng làm tương tự.

a) Thuật toán cho công thức R-K4 cổ điển

Input: $f, a, b, y0, h$

Output: Giá trị nghiệm y_i của phương trình giải bằng R-K tại các điểm $x_i = a + ih$

Sơ đồ thuật toán cho công thức R-K4, độ phức tạp của nó là tuyến tính:



<u>*Nhận xét:*</u> Thuật toán các công thức R-K khác nhau chỉ khác nhau ở bước tính các giá trị k_i. Để không cần tạo quá nhiều gói (mỗi gói cho một công thức cụ thể) trong khi viết chương trình thì có thể lưu trữ giá trị của bảng Butcher và khi muốn áp dụng công thức nào thì ta chỉ cần lấy bộ hệ số đã lưu và thay vào một gói tính R-K duy nhất.

b) Thuật toán chương trình dùng R-K giải hệ ODEs cấp 1

Ở đây lấy ví dụ là sơ đồ của thuật toán giải hệ 2 ODEs bằng RK4 cổ điển, các trường hợp khác tương tự (hệ nhiều hơn 2 phương trình, sử dụng công thức khác RK4 cổ điển,...). Đối với phương trình cấp cao trong một chiều cũng có thể đưa về hệ k phương trình cấp 1 nên các thuật toán là tương tự.

Xét hệ: $\begin{cases} y' = f(x,y,z) \\ z' = g(x,y,z) \end{cases}$ $\quad\begin{matrix} x \in [a,b] \\ y(a) = y0, z(a) = z0 \end{matrix}$

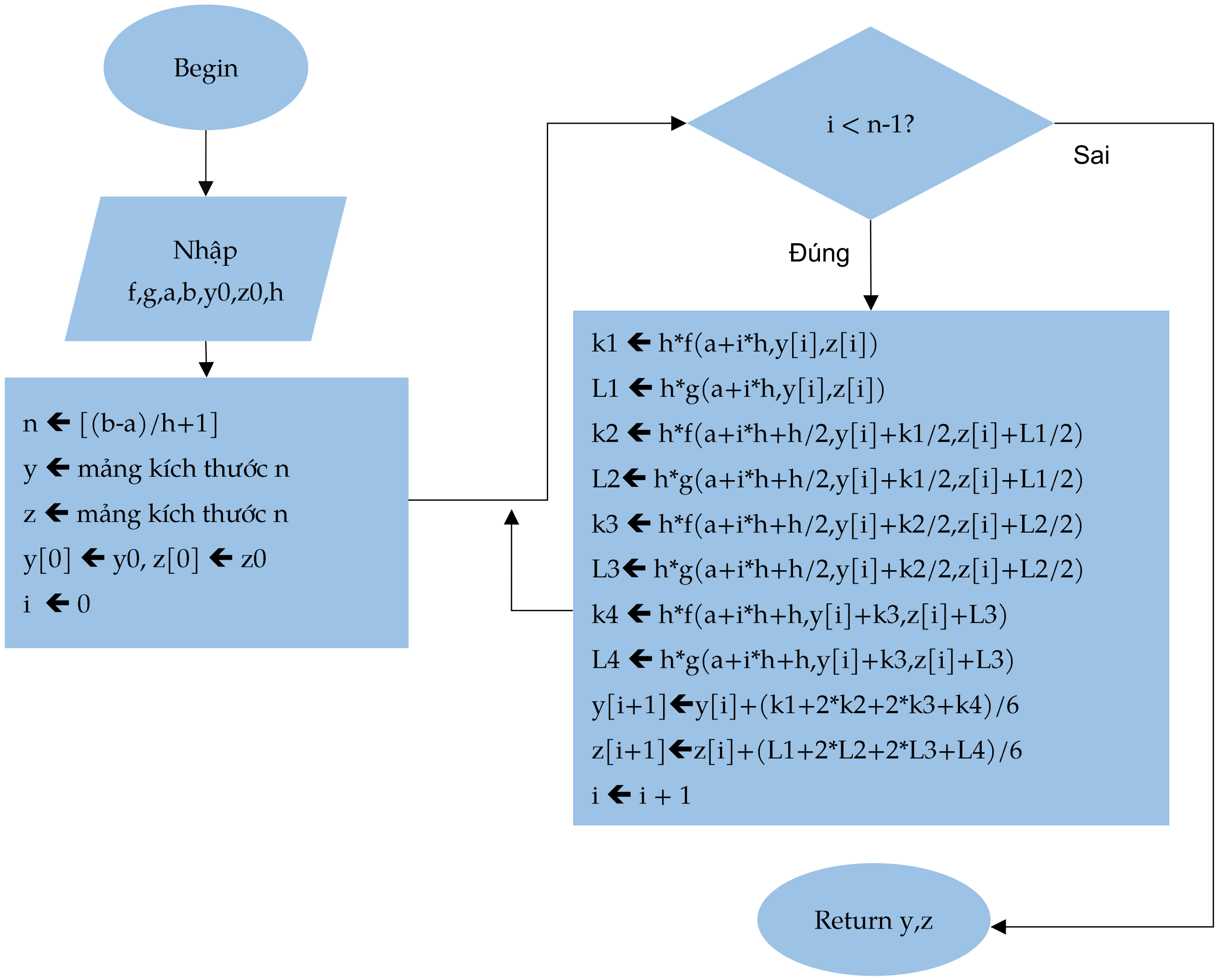
Input: f,g,a,b,y0,z0,h

- *f,g:* Là hàm f, g trong hệ trên(nếu biểu diễn thuật toán giải phương trình cấp 2 có thể mặc định $f = z$ và chỉ input g).

- *a,b,y0,z0:* Lần lượt là điểm bắt đầu, kết thúc, giá trị ban đầu như hệ nêu trên.

- *h*: bước lưới (lưới đều)

Output: y,z là mảng chứa nghiệm của hệ tại các x_i



<u>*Ghi chú:*</u> Mảng a kích thước n là dãy $a_0, a_1, \dots a_{n-1}$, ký hiệu $a[i] \coloneqq a_i$. Nếu tính toán y,z như các vector thì sẽ có cách biểu diễn ngắn gọn hơn.

c) Thuật toán R-K thích ứng (Runge-Kutta-Fehlberg)

Trong thuật toán này, h sẽ được thay đổi sao cho sai số đạt được là tốt nhất theo yêu cầu mà vẫn đảm bảo h nằm trong khoảng cho phép, hệ số được lấy từ bảng Butcher ở mục 8 phần B. Sau đây là thuật toán R-K-F cho ODE cấp 1 trong 1 chiều.

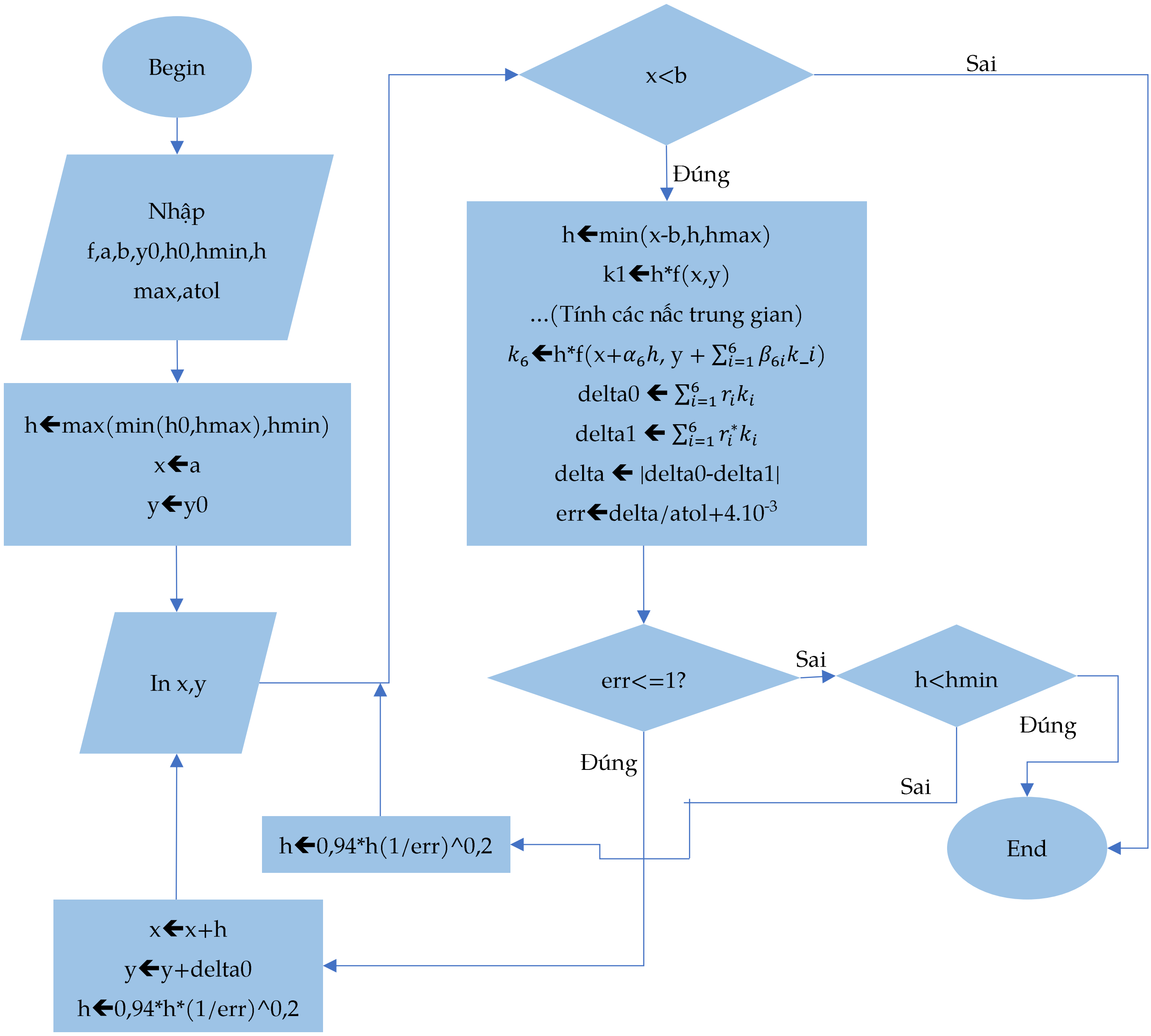
Input: $f, a, b, y0, h0, hmin, hmax, atol$

- *f,a,b,y0:* Giống như thuật toán giải phương trình cấp 1

- *h0,hmin,hmax:* Bước h bắt đầu, h nhỏ nhất và lớn nhất cho phép khi thay đổi h

- *atol:* Sai số mong muốn.

Output: Bộ các giá trị x_i, y_i

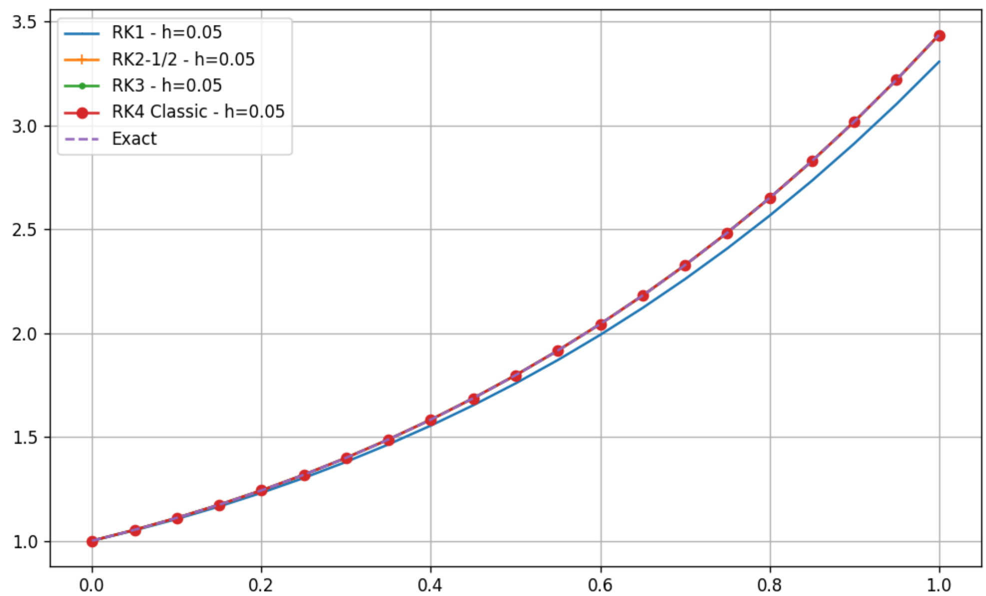


<u>*Ghi chú:*</u> Trong thuật toán này, $atol$ là sai số mong muốn. Trong bước tính $err = \frac{delta}{atol} + 4.10^{-3}$ có cộng một lượng nhỏ để đảm bảo err khác 0 do sau đó có thực hiện phép chia cho err ở bước thay đổi h và do $err \geq 4e - 3$ nên cũng giới hạn tăng h lớn nhất là 3 lần. Bước so sánh $h < hmin$ để thuật toán không chạy quá lâu, khi h quá bé vẫn có thể sẽ tiếp tục giảm h, cho dù có h có tăng do sai số đạt yêu cầu thì tỉ lệ h tăng vẫn sẽ ít (lớn nhất là gấp 3 lần), bước h vẫn nhỏ, mà khi đó thì x không thể đi tới b (hoặc có thể tới nhưng sau rất nhiều vòng lặp). Trong công thức thay đổi h có hệ số 0.94 (có thể lấy trong khoảng 0.8-0.95)để đảm bảo an toàn sai số.

2. Thử nghiệm và đánh giá

VD1: Với phương trình: $y' = x + y; y(0) = 1; 0 \leq x \leq 1; h = 0.05$

Phương trình này có nghiệm chính xác là $y = 2e^x - x - 1$, dưới đây là đồ thị được vẽ từ các giá trị của y được trả về từ các công thức R-K khác nhau.

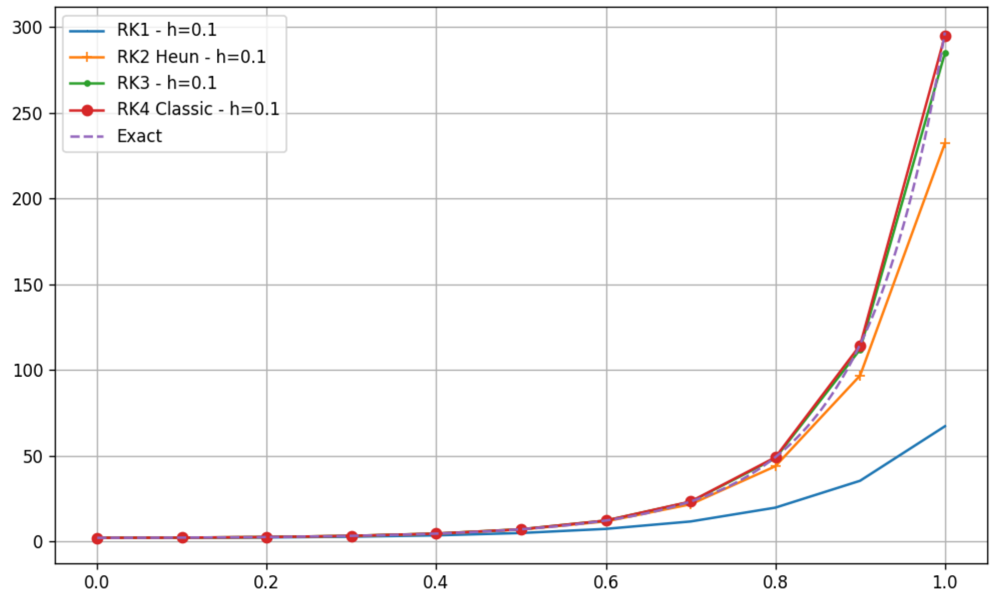


<u>*Nhận xét*</u>: Nghiệm giải ra rất giống với nghiệm đúng đối với các công thức R-K có bậc lớn hơn 1 (gần như đè lên nhau và gần với nghiệm đúng), chỉ có R-K1 là có sai số khá lớn, công thức này *có sai số rất nhỏ ở những điểm gần điểm bắt đầu, càng ra xa thì sai số càng lớn*. Tuy nhiên ta chưa nhìn rõ được sự chênh lệch này cho nên xét VD2.

VD2: Giải phương trình: $y' = 10xy, \quad y(0) = 2, \quad 0 \leq x \leq 1, \quad h = 0.1$

Phương trình này có nghiệm đúng là $y(x) = 2e^{5x^2}$

Dưới đây là đồ thị nghiệm thu được:



<u>*Nhận xét:*</u> Ở ví dụ này ta đã thấy rõ hơn được đánh giá từ ví dụ 1, về cơ bản thì dáng điệu của hàm đã đúng, *khi sử dụng công thức bậc cao hơn thì kết quả có độ chênh lệch so với nghiệm đúng là ít hơn*, như ví dụ này thì RK1 lệch rất nhiều, độ lệch RK2 < RK3 < RK4 và RK4 gần như là chuẩn xác, công thức RK4 có sai số tốt.

VD3: Thông thường thì giá trị tính tại điểm càng xa thì sai số càng lớn, nhưng vẫn có trường hợp không như vậy. Xét bài toán:

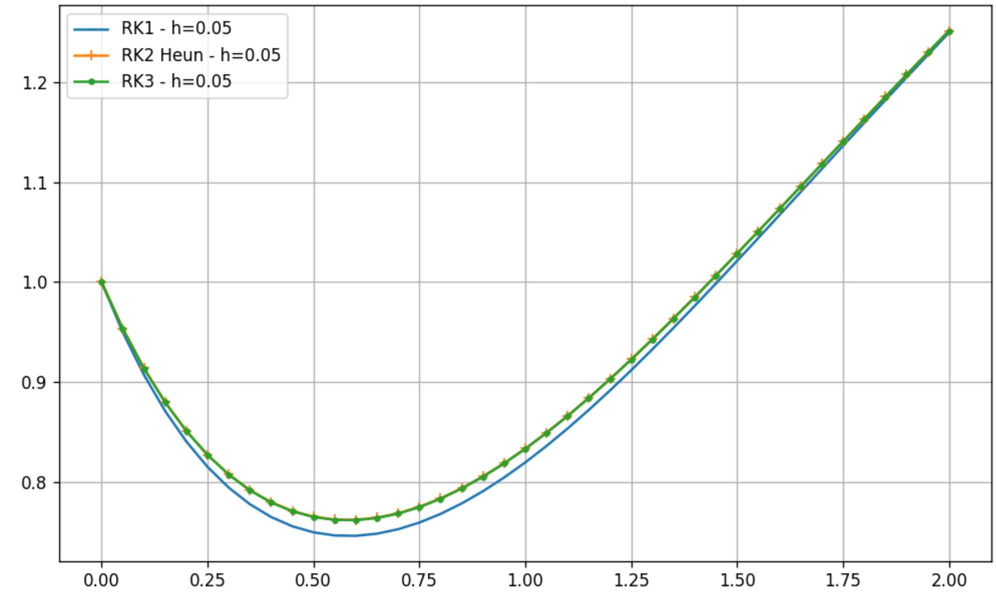
$$y' = x - y^2, 0 \leq x \leq 2, y(0) = 1, h = 0.05$$

Đây là một phương trình phi tuyến, không giải được bằng cách bình thường cho nên ta cần giải số, biết $y(2) \approx 1.2513, y(0.6) \approx 0.76209$. Giá trị trả về bởi các công thức:

- RK1 (Euler): $y(2.0) \approx 1.24934$; $y(0.6) \approx 0.746447$
- RK2 (Heun): $y(2.0) \approx 1.25134$; $y(0.6) \approx 0.762501$
- RK3 (Simpson): $y(2.0) \approx 1.25131$; $y(0.6) \approx 0.762085$

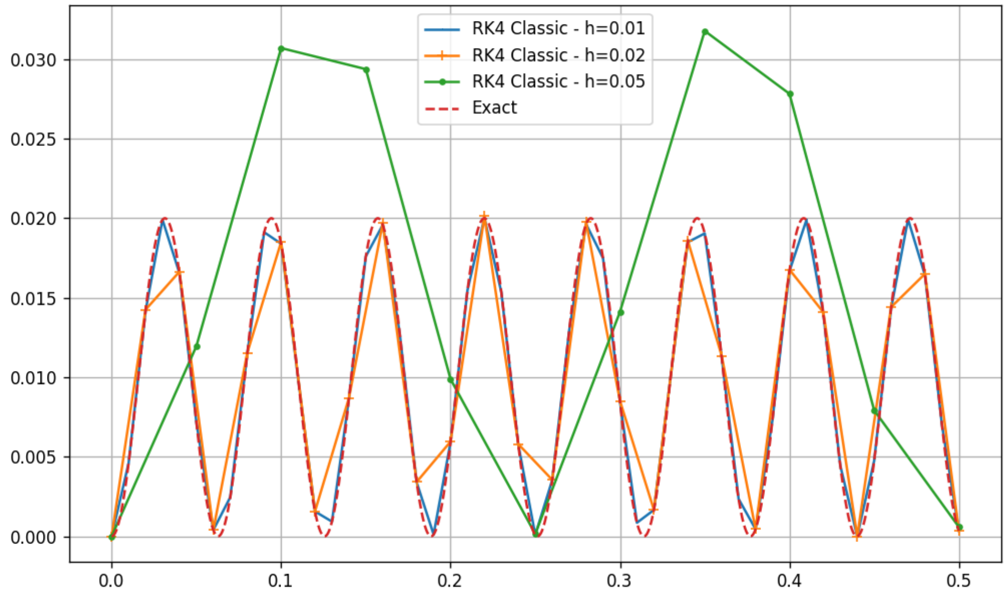
<u>*Nhận xét:*</u> Các công thức đều trả về nghiệm khá chính xác và gần như trùng nhau, ở ví dụ này thì ta thấy có một điểm đặc biệt đó là *giá trị sai số của công thức tại điểm giữa lại có thể lớn hơn điểm ở xa nhất so với điểm đầu*. Trường hợp này khá đặc biệt, tuy nhiên thông thường thì các điểm ở xa thì sẽ có sai số lớn hơn.

Đồ thị nghiệm thu được qua các công thức RK:



VD4: Phương trình $y' = sin(100x);\ 0 \le x \le 0.5;\ y(0) = 0; h = 0.01, 0.02, 0.05$

Nghiệm đúng $y(x) = -\frac{cos(100x)}{100} + \frac{1}{100}$. Đồ thị nghiệm thu được:

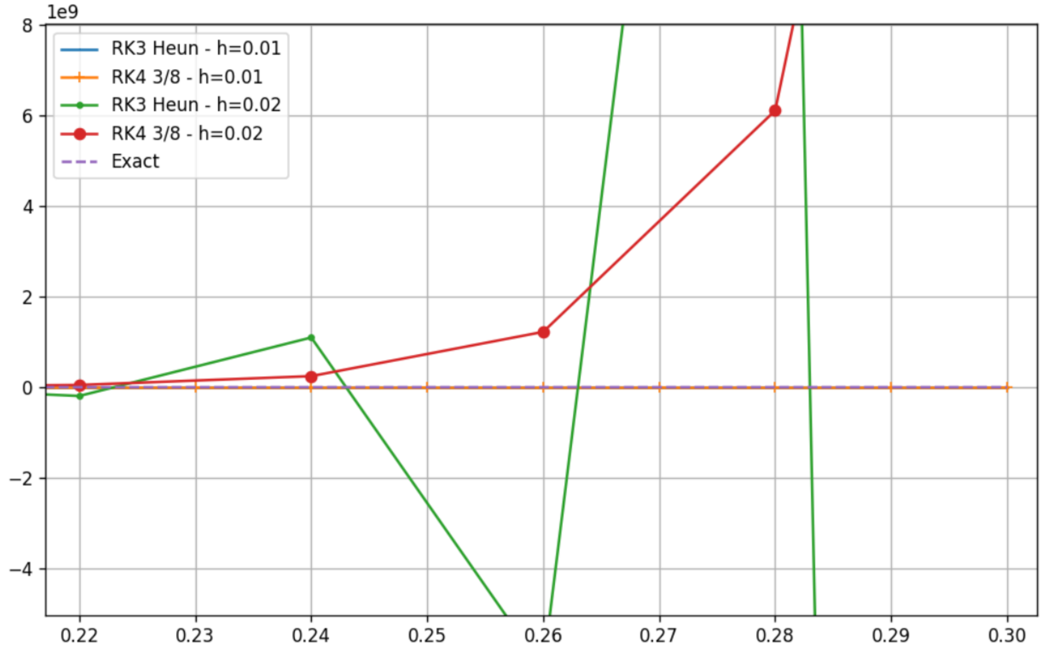


<u>*Nhận xét:*</u> Tuy sử dụng cùng một công thức RK4 nhưng khi $h = 0.05$ nghiệm lệch đi khá nhiều còn với h bé hơn với cùng công thức RK4 thì lại cho kết quả gần với đồ thị đúng hơn. Như vậy, *với h nhỏ thì kết quả sẽ đáng tin cậy hơn, sai số sẽ tốt hơn.* Điều này cũng có nghĩa là khối lượng tính toán sẽ tăng lên.

VD5: Sự mất ổn định. Xét phương trình:

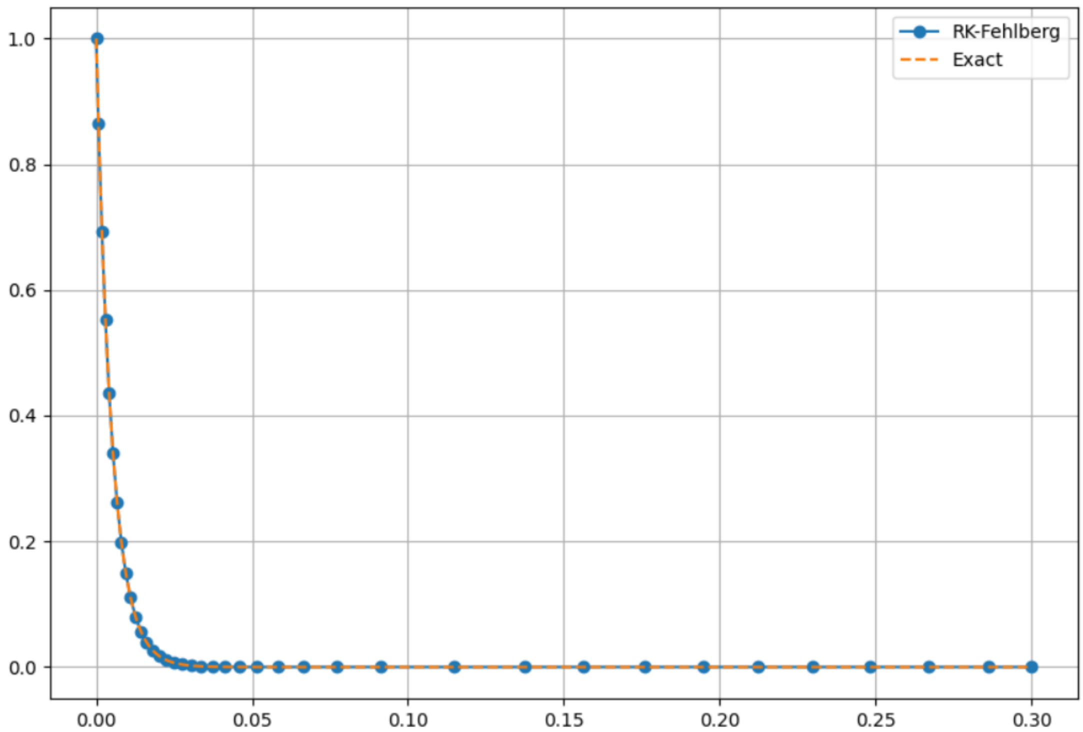
$$y' = -200y;\ y(0) = 1;\ 0 \le x \le 0.3;\ h = 0.01, 0.02$$

Đồ thị nghiệm thu được:



<u>*Nhận xét:*</u> Với giá trị h nhỏ và công thức sai số RK3, RK4 là công thức cho chính xác tốt, ta hi vọng sẽ có sai số tốt tuy nhiên ở ví dụ này thì khi $h = 0.02$ đã khá nhỏ nhưng do nó *nằm ngoài miền ổn định* của công thức nên nghiệm đã không hội tụ, ngược lại với $h = 0.01$ thì nghiệm đã hội tụ vì *nằm trong miền ổn định*.

VD6: Dùng RKF cho ví dụ 5 với $h0 = 0.1, h_{min} = 10^{-4}, h_{max} = 0.2, atol = 10^{-6}$:



<u>*Nhận xét:*</u> Với thuật toán bước thay đổi, phương pháp cho nghiệm hội tụ về nghiệm đúng mặc dù h0=0.1 nằm ngoài miền ổn định của công thức. Số điểm cần tính (vòng lặp) cũng không lớn hơn quá nhiều so với khi dùng thuật toán bước cố định, kết quả nhận được thì lại rất tốt.

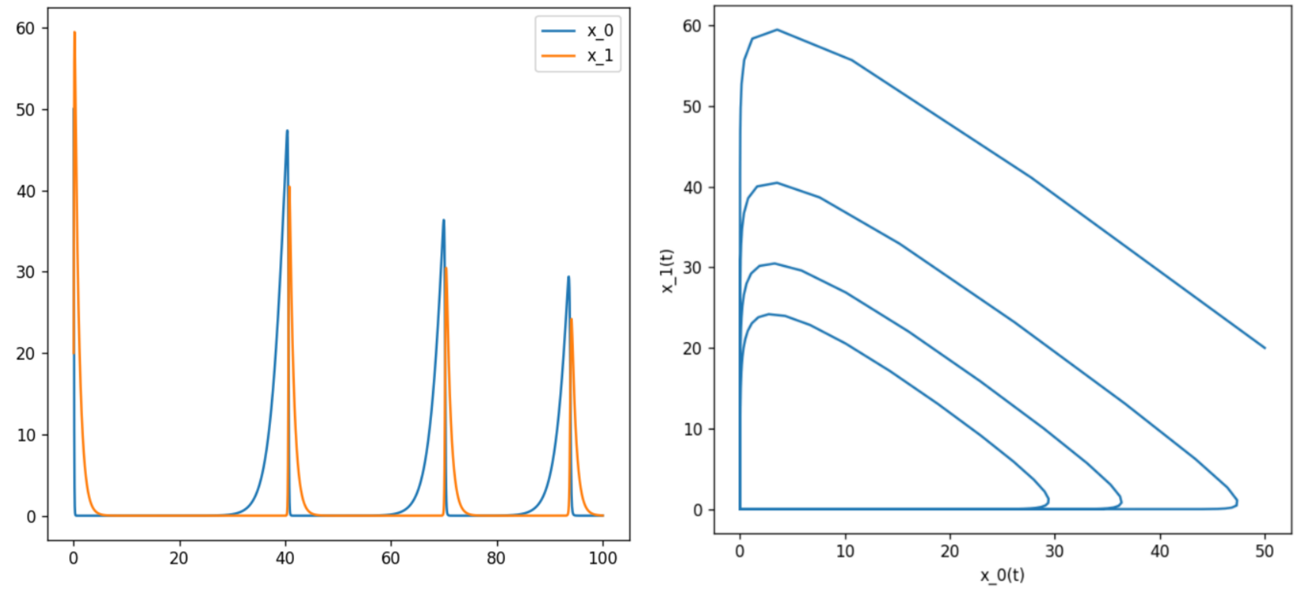
VD7: Thử nghiệm với một hệ phương trình bằng thuật toán giải hệ phương trình được nêu ở trên với công thức R-K4 cổ điển.

Xét hệ phương trình (hệ thú mồi Lotka-Voltera):

$$\begin{cases} n' = 0.6n\left(1 - \dfrac{n}{100}\right) - 0.4np \\ p' = -1.2p + 0.4np \end{cases}$$

$$n(0) = 50, p(0) = 20, ; 0 \le t \le 100, ; h = 0.05$$

trong đó $n(t)$ là số lượng loài thỏ, $p(t)$ là số lượng loài cáo tại thời điểm t



Mô hình Lotka-Volterra(x_0 là hàm n, x_1 là hàm p)

<u>*Nhận xét:*</u> Đây là hệ khó giải bằng thông thường, qua lời giải số ta thấy được dáng điệu của hàm và sự thay đổi của đồ thị sẽ mô phỏng được phần nào quy luật tự nhiên.

**Đối với phương trình cấp cao và hệ nhiều phương trình hơn hoàn toàn tương tự, tải chương trình và chạy thử ở phần phụ lục.*

D. KẾT LUẬN

Qua các phần trên, ta rút ra được một số kết luận về phương pháp R-K, cụ thể là các công thức R-K hiện. Đó là thuật toán dễ triển khai, tính toán đã được giảm bớt so với việc phải tính nhiều đạo hàm tại 1 điểm như Taylor, sai số của công thức hay dùng như R-K4 cũng đã đủ tốt trong nhiều trường hợp. Các công thức R-K bậc cao vẫn cần lượng tính toán khá lớn và vẫn còn sự mất ổn định. Các công thức R-K đã cho ta được dáng điệu của hàm và tính gần đúng được kết quả như ta mong muốn, R-K bậc cao không còn là lời giải thô như Euler hiện, những phương pháp thuộc họ R-K được phát triển để giải cho những dạng bài toán khác nhau, phức tạp hơn. Cũng có phương pháp ẩn hoặc phương pháp đa bước để giảm khối lượng tính toán nhưng vẫn đạt sai số mong muốn và có sự ổn định hơn. Nhìn chung, phương pháp R-K cho ta một kết quả tốt, trong khi tính toán cần tránh những lỗi chọn h để đạt được kết quả tốt nhất. Phương pháp R-K có thể được dùng để mô phỏng các hệ thống trong kỹ thuật…

E. TÀI LIỆU THAM KHẢO

[1] *J. C. Butcher* - Numerical Methods for Ordinary Differential Equations-J. Wiley (2003)

[2] *Phạm Kỳ Anh*, Giải tích số, NXB Đại học Quốc gia Hà Nội 1996, trang 145-156.

[3] *Lê Trọng Vinh*, Giải tích số, NXB , trang 197-215,219-222.

[4] *Lê Trọng Vinh, Trần Minh Toàn*, Giáo trình phương pháp tính và Matlab, NXB Bách khoa, trang 205-220

[5] *Trần Văn Trản*, Phương pháp số thực hành tập 1, NXB Đại học Quốc gia Hà Nội, 2007, trang 380-388

[6] *J Dougles Faires Ric, Burden* , Numerical Method (third edition),2002.

F. PHỤ LỤC

Code của các chương trình thực hiện thuật toán trong bài báo cáo này được kèm theo hoặc có tại địa chỉ: https://anhnb206110.github.io/Runge_Kutta/

Hướng dẫn sử dụng có được viết ở file HDSD.html