CHƯƠNG 6
GIẢI GẦN ĐÚNG PHƯƠNG TRÌNH VI PHÂN

MỤC ĐÍCH, YÊU CẦU

Sau khi học xong chương 3, yêu cầu sinh viên:

1. Hiểu được vai trò và tầm quan trọng của bài toán giải gần đúng phương trình vi phân.
2. Nắm được các phương pháp tìm nghiệm gần đúng của phương trình vi phân.
3. Biết cách áp dụng các phương pháp trên vào việc giải quyết các bài toán thực tế.
4. Biết cách đánh giá sai số của từng phương pháp.

6.1. MỞ ĐẦU

Nhiều bài toán khoa học kỹ thuật dẫn về việc tìm nghiệm phương trình vi phân thỏa mãn một số điều kiện nào đó. Những phương trình vi phân mô tả những hệ cơ học, lý học, hóa học, sinh học nói chung rất phức tạp, không hy vọng tìm lời giải đúng.

Trong chương này ta nghiên cứu bài toán đơn giản nhất của phương trình vi phân là ***bài toán Cauchy đối với phương trình vi phân cấp 1*** như sau:

Hãy tìm hàm $y=y(x)$ thỏa mãn:

$$y'(x) = f(x,y) \quad x \in [a,b], \; x_0 = a \tag{6.1}$$

$$y(x_0) = y_0 \tag{6.1b}$$

Điều kiện (6.1b) được gọi là điều kiện ban đầu hay điều kiện Cauchy.

Tương tự, bài toán Cauchy đối với phương trình vi phân cấp n được mô tả như sau:

Hãy tìm hàm $y=y(x)$ thỏa mãn:

$$y^{(n)} = f(x,y,y',y^{(2)},...,y^{(n-1)}) \tag{6.2}$$

$$y(x_0) = \alpha_0, \; y'(x_0) = \alpha_1, \; y^{(2)}(x_0) = \alpha_2, \; ..., \; y^{(n-1)}(x_0) = \alpha_{n-1}$$

trong đó $f()$ là một hàm đã biết của n+1 đối số $x,y,y',y^{(2)},...,y^{(n-1)}$; x_0, b, α_0, α_1 ..., α_{n-1} là những số cho trước.

(6.1) còn được mở rộng cho hệ thống các phương trình vi phân cấp một với bài toán Cauchy chư sau:

$$\begin{aligned} y_1' &= f_1(x,y_1, y_2,..., y_n) \\ y_2' &= f_2(x,y_1, y_2,..., y_n) \\ &\ldots \\ y_n' &= f_n(x,y_1, y_2,..., y_n) \end{aligned} \tag{6.3}$$

$y_1(x_0) = \alpha_1, y_2(x_0) = \alpha_2, \ldots, y_n(x_0) = \alpha_n$

$x \in [a,b], x_0 = a$

Nếu đặt

$\vec{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$

$\vec{y} = [y_1, y_2, \ldots, y_n]^T$

$\vec{y} = [y'_1, y'_2, \ldots, y'_n]^T$

$\vec{f} = [f_1, f_2, \ldots, f_n]^T$

Bài toán (6.3) có thể viết gọn hơn dưới dạng vectơ như sau:

$\vec{y} = \vec{f}(x, \vec{y}), \quad x \in [a,b], \quad x_0 = a$

$\vec{y}(x_0) = \vec{\alpha}$

Ghi chú. Phương trình vi phân cấp n có thể đưa về hệ các phương trình vi phân cấp một bằng phép biến đổi

$y_1 = y, y_2 = y', \ldots, y_i = y^{(i-1)}, \ldots, y_n = y^{(n-1)}$

Nói chung có hai nhóm phương pháp để giải các phương trình vi phân thường:

Phương pháp tìm nghiệm chính xác: bằng cách dựa vào cách tính tích phân trực tiếp, xác định được dạng tổng quát của nghiệm rồi dựa vào điều kiện ban đầu để xác định nghiệm riêng cần tìm.

Phương pháp tìm nghiệm gần đúng xuất phát từ điều kiện ban đầu. Phương pháp này có thể áp dụng cho một lớp phương trình vi phân rộng hơn rất nhiều so với phương pháp trực tiếp, do đó được dùng nhiều trong thực tế.

Trong phần tiếp theo ta sẽ tập trung vào nhóm phương pháp thứ hai.

6.2. PHƯƠNG PHÁP EULER

Trở lại bài toán

$$y'(x) = f(x,y) \quad x \in [a,b], \quad x_0 = a \qquad (6.4)$$

$y(x_0) = y_0$

Cách giải gần đúng (6.4) là tìm các giá trị gần đúng y_i của giá trị đúng $y(x_i)$ tại các điểm x_i, i = 0,1,2,... n, trong đó

$a = x_0 < x_1 < \ldots < x_n = b$

$x_i = x_0 + ih, \quad i = 0,1,\ldots,n-1$

$$h = \frac{b-a}{n}$$

Ta đã biết $y_0 = \alpha_0$, ta sẽ lần lượt xác định y_1 tại x_1, rồi y_2 tại x_2, và nói chung từ giá trị gần đúng y_i tại x_i ta sẽ tính y_{i+1} tại x_{i+1}.

Phương pháp Euler cũng như một vài phương pháp sẽ được trình bày sẽ dựa vào giả thiết sau đây (cho dù giả thiết này nói chung không thể kiểm tra được)

Giả thiết rằng bài toán (6.4) có nghiệm duy nhất $y = y(x)$, $x \in [a,b]$, $a = x_0$, và nghiệm $y(x)$ đủ trơn, nghĩa là nó có đạo hàm đến cấp đủ cao. (6.5)

Ta khai triển Taylo nghiệm y(x) của (6.4) tại x_i

$$y(x) = y(x_i) + \frac{x - x_i}{1!} y'(x_i) + \frac{(x - x_i)^2}{2!} y''(c_i), \; c_i \in (x_i, x) \qquad (6.5)$$

Thay $x = x_{i+1} = x_i + h$, $y'(x_i) = f(x_i, y(x_i))$ vào đẳng thức trên ta có

$$y(x_{i+1}) = y(x_i) + h\, f(x_i, y(x_i)) + \frac{h^2}{2} y''(c_i), \; c_i \in (x_i, x_{i+1}) \qquad (6.6)$$

Bỏ qua số hạng cuối cùng bên phải, đồng thời thay các giá trị đúng $y(x_{i+1})$, $y(x_i)$, $f(x, y(x_i))$ bằng các giá trị xấp xỉ y_{i+1}, y_i, $f(x, y_i)$ vào (6.6) ta có

$$y_{i+1} = y_i + h\, f(x_i, y_i) \qquad (6.7)$$

Với giá trị $y_0 = y(x_0) = \alpha_0$ ban đầu (như vậy y_0 là giá trị đúng của $y(x_0)$), ta có thể tính tiếp các giá trị y_i, i =1,2, ..., n.

Công thức (6.7) được gọi là công thức Euler. Công thức này cho ta cách tính y_{i+1} khi đã biết y_i mà không phải giải một phương trình nào. Vì vậy phương pháp Euler là một phương pháp hiện.

Sai số địa phương của phương pháp Euler là

$$R_i(h) = y(x_i) - y_i = \frac{h^2}{2!} y''(c_{i-1}) = O(h^2) \qquad (6.8)$$

Người ta chứng minh được rằng: sai số của phương pháp Euler tại điểm x_i là

$$| y_i - y(x_i) | \le Mh \qquad (6.9)$$

trong đó M là hằng số không phụ thuộc h.

Điều này chứng tỏ rằng khi $h \to 0$ thì $y_i \to y(x_i)$ tại mọi điểm x_i cố định. Tuy nhiên việc xác định giá trị M rất phức tạp. Vì vậy trong thực hành người ta thường làm như sau: Quá trình tính được chia làm nhiều bước, khoảng cách giữa các điểm x_i và x_{i+1} ở bước sau sẽ là một nửa khoảng cách của bước trước, tức là $h_{k+1} = h_k/2$. Nếu gọi lần đầu tiên với $h = \frac{b-a}{n}$ ta gọi là lần thứ 0 với n+1 điểm chia thì tại lần thứ k+1 sẽ có $n.2^{k+1} + 1$ điểm chia. Trong số điểm chia này thì có $n.2^k + 1$ điểm chia trùng với lần thứ k. Các điểm trùng đó là 0,2,4,..., $n.2^{k+1}$. Tại các điểm chia này ta có ở lần thứ k giá trị xấp xỉ của y_i là $y_i^{(k)}$, còn ở lần thứ k+1 thì giá trị xấp xỉ giá trị y tại vị trí này lại là $y_{2i}^{(k+1)}$. Ta xét đại lượng sau

$$\text{maxdiff} = \max_i | y_{2i}^{(k+1)} - y_i^{(k)} |, \; i = 0,1,2,...,n.2^k$$

Và sẽ dừng quá trình tính toán nếu maxdiff nhỏ hơn giá trị ε khá nhỏ cho trước.

Ta có thể mô tả thuật toán Euler để cài đặt trên máy tính theo các bước sau:

a. Thuật toán cho một lần chia khoảng duy nhất.

Nhập a, b , y_0 và n.

Đặt $h = \frac{b-a}{n}$, $x_0 = a$.

Với i =1,2,..., n tính

$y_i = y_{i-1} + hf(x_{i-1},y_{i-1})$

$x_i = x_{i-1} + h$

b.Thuật toán cho nhiều lần chia khoảng.

- **Bước 0:** Nhập a, b , y_0 , kmax và $\varepsilon>0$.

Đặt $h_0 = \frac{b-a}{n}$, $x_0^{(0)} = a$, $y_0^{(0)} = y_0$

Tính $y_i^{(0)} = y_{i-1}^{(0)} + hf(x_{i-1}^{(0)}, y_{i-1}^{(0)}), x_i^{(0)} = x_{i-1}^{(0)} + h_0$, i = 1,2,...,n

- **Bước 1:**

Đặt $h_1 = \frac{h_0}{2}$, $x_0^{(1)} = a$, $y_0^{(1)} = y_0$

Tính $y_i^{(1)} = y_{i-1}^{(1)} + hf(x_{i-1}^{(1)}, y_{i-1}^{(1)}), x_i^{(1)} = x_{i-1}^{(1)} + h_1$, i = 1,2,...,n.2

$d_1 = \max_i | y_{2i}^{(1)} - y_i^{(0)} |$, i=0,1,2,...,n

Nếu $d_1 <\varepsilon$ thì dừng thuật toán và lấy mẫu (x_0,y_0), (x_1,y_1), ..., (x_{2n},y_{2n}) làm nghiệm xấp xỉ.

Nếu 2 điều trên đây không xẩy ra thì chuyển qua bước (3).

...

- **Bước k:**

Đặt $h_k = \frac{h_{k-1}}{2}$, $x_0^{(k)} = a$, $y_0^{(k)} = y_0$

Tính $y_i^{(k)} = y_{i-1}^{(k)} + hf(x_{i-1}^{(k)}, y_{i-1}^{(k)}), x_i^{(k)} = x_{i-1}^{(k)} + h_k$, i = 1,2,...,n.2^k

$d_k = \max_i | y_{2i}^{(k)} - y_i^{(k-1)} |$, i=0,1,2,...,n.$2^{k-1}$

Nếu $d_k <\varepsilon$ thì dừng thuật toán và lấy mẫu (x_0,y_0), (x_1,y_1), ..., (x_N,y_N), trong đó $N = n.2^k$ làm nghiệm xấp xỉ.

Nếu $k \geq kmax$ thì thông báo phép lặp chưa hội tụ và cũng dừng thuật toán.

Nếu 2 điều trên đây không xẩy ra thì chuyển qua bước (k+1).

c.Chương trình cài đặt thuật toán Euler:

```
//EULER.CPP
/*Phuong phap Euler giai gan dung phuong trinh vi phan y'=f(x,y)*/
#include "zheader.cpp"
#define kmax 17
const double epsi=1.0E-02;
double g(double,double);//a=0,b=1
double yg(double);//La nghiem dung y=y(x) cua phuong trinh y'=g(x,y)
```

```
int euler(double (*f)(double,double),double (*yf)(double));
/*Phuong phap Euler giai phuong trinh vi phan tren [a,b].
  y'(x)=f(x,y), y(x0)=y0.
  Neu buoc lap vuot qua kmax thi tra ve false
  Ham yf la nghiem dung de so sanh*/
//=================================================
double g(double x,double y)
 {return x*y/2;
 }
//=================================================
double yg(double x)
 {return exp(x*x/4);
 }
//=================================================
//Ham de tim y(x).
int euler(double (*f)(double,double),double (*yf)(double))
 {clrscr();
  double a,b,h,maxdiff,x[kmax],y[kmax],y1[kmax];int n,m,i;
  cout<<endl<<"Can duoi: ";cin>>a;
  cout<<"Can tren: ";cin>>b;
  cout<<"Dieu kien ban dau: y(a) = ";cin>>y[0];
  n=1;
  h=b-a;
  x[0]=a;
  y[1]=y[0]+h*f(x[0],y[0]);
  do
   {for(i=0;i<=n;i++) y1[i]=y[i];//Gan y1=y de bat dau tinh y moi
    m=n;
    n=n*2;
    h=h/2;
    for(i=0;i<n;i++)
     {x[i+1]=x[i]+h;
      y[i+1]=y[i]+h*f(x[i],y[i]);
     }
    maxdiff=0;
```

```
 for(i=0;i<=m;i++)
  maxdiff=maxdiff<fabs(y[2*i]-y1[i])?fabs(y[2*i]-y1[i]):maxdiff;
 if(maxdiff>epsi&&2*n>kmax-1)
  {cout<<endl<<"Chua hoi tu voi  "<<n<<"  khoang chia";
   delay(1000);break;
  }
 }
while(maxdiff>epsi);

cout<<endl<<" i    x[i]     y[i]       y(x[i])";
for(i=0;i<=n;i++)
 {cout<<endl<<setw(4)<<i;
  cout<<setw(10)<<setprecision(2)<<x[i];
  cout<<setw(10)<<setprecision(4)<<y[i];
  cout<<setw(10)<<setprecision(4)<<yf(x[i]);
 }
cout<<endl<<endl<<"So khoang chia :";
cout<<setw(10)<<n;
cout<<endl<<"Sai so |y[i]-y(x[i])| cuc dai:";
cout<<setw(10)<<setprecision(4)<<maxdiff;
//================Ghi vao tep EULER.DAT
FILE *fp;
fp=fopen("EULER.DAT","wt");
fprintf(fp," i    x[i]     y[i]       y(x[i])");
for(i=0;i<=n;i++)
 {fprintf(fp,"\n%4d",i);
  fprintf(fp,"%10.2f",x[i]);
  fprintf(fp,"%10.4f",y[i]);
  fprintf(fp,"%10.4f",yf(x[i]));
 }
fprintf(fp,"\n\nSo khoang chia : %4d",n);
fprintf(fp,"\nSai so |y[i]-y(x[i])| cuc dai: %10.4f",maxdiff);
fclose(fp);
getch();
clrscr();
```

```
  if(n>kmax) return false;
   else  return true;
 }
//================================================
void main()
 {char mchon;
  while(true)
   {clrscr();
  gotoxy(20,2);cout<<"PHUONG PHAP EULER";
  gotoxy(2,2);
  cout<<endl<<" 1. Giai phuong trinh vi phan ";
  cout<<endl<<" z. Ket thuc";
  cout<<endl<<" Nhan phim  1 -> z de chon";
  mchon=toupper(getch());
  if(mchon=='Z') break;
  switch(mchon)
   {case '1':clrscr();
             euler(g,yg);
             break;
   }
  gotoxy(2,22);
  cout<<"Nhan phim bat ky de tiep tuc";
  getch();
  }
}
```

Nhận xét.

Ưu điểm của phương pháp Euler là tính toán đơn giản, nhưng nhược điểm là độ chính xác thấp. Khi ta giảm bước h thì độ chính xác tăng lên và sai số tích lũy cũng tăng lên. Để giảm sai số tích luỹ, trong chương trình tính toán trên máy tính ta nên dùng các đại lượng có độ chính xác gấp đôi.

Phương pháp Eurler có thể áp dụng cho hệ thống phương trình vi phân cấp một (6.3).

Sử dụng các ký hiệu như trong (6.3) ta có công thức Euler giải bài toán Cauchy có dạng sau:

$$\vec{y}_0 = \vec{y}(x_0) = \alpha$$
$$\vec{y}_{i+1} = \vec{y}_i + h\,\vec{f}(x_i,\vec{y}_i),\ i=0,1,2,...,n-1 \qquad (6.10)$$

Đối với hệ thống hai phương trình vi phân cấp một:

$y' = f_1(x,y,z)$

$z' = f_2(x,y,z)$

với điều kiện ban đầu:

$y(x_0) = \alpha_1$, $z(x_0) = \alpha_2$

Công thức (6.8) có dạng sau:

$y_0 = y(x_0) = \alpha_1$, $z_0 = z(x_0) = \alpha_2$

$y_{i+1} = y_i + hf_1(x_i, y_i, z_i)$

$z_{i+1} = z_i + hf_2(x_i, y_i, z_i)$ i=0,1,...,n-1 (6.11)

Ví dụ.

Cho hệ thống phương trình vi phân cấp một:

$y' = (z-y)x$

$z' = (z+y)x$

Với điều kiện ban đầu: y(0)=z(0)=1.

Hãy tìm nghiệm gần đúng bằng phương pháp Euler trên khoảng [0,0.6] với bước h=0.1.

Giải:

Ta có $x_i = 0.1i$, i=0,1,2,...,6.

Xuất phát từ y(0)=z(0)=1, áp dụng (6.11) ta nhận được kết quả tính toán sau:

i	x_i	y_i	$f_1(x_i, y_i, z_i)$	$hf_1(x_i, y_i, z_i)$	z_i	$f_2(x_i, y_i, z_i)$	$hf_2(x_i, y_i, z_i)$
0	0	1.0000	0	0	1.0000	0	0
1	0.1	1.0000	0	0	1.0000	0.2000	0.0200
2	0.2	1.0000	0.0040	0.0004	1.0200	0.4040	0.0404
3	0.3	1.0004	0.0180	0.0018	1.0604	0.6182	0.0618
4	0.4	1.0022	0.0480	0.0048	1.1222	0.8498	0.0850
5	0.5	1.0070	0.1001	0.0100	1.2072	1.1071	0.1107
6	0.6	1.0170			1.3179		

6.3.PHƯƠNG PHÁP EULER CẢI TIẾN

Để tăng độ chính xác của phương pháp Euler người ta làm như sau:

Theo công thức Newton-Lepnitz, ta có

$$y(x2) - y(x1) = \int_{x1}^{x2} y'(x)dx$$

Tính gần đúng tích phân xác định ở vế phải bằng công thức hình thang ta có:

$$\int_{x1}^{x2} y'(x)dx \approx \frac{h}{2} [y'(x1)+y'(x2)] = \frac{h}{2} [f(x1,y(x1)) + f(x2,y(x2)], \text{ trong đó } h=x2-x1$$

Thay x1 bằng x_i, x2 bằng x_{i+1}, ta có

$$y_{i+1} = y_i + \frac{h}{2}\,[f(x_i,y_i) + f(x_{i+1},y_{i+1})]\ ,\ i=0,1,...,n-1 \qquad (6.12)$$

với $y_0 = \alpha$.

Người ta đã chứng minh được rằng sai số của (6.12) tại điểm x_i là:

$$| y_i - y(x_i) | \le Mh^2$$

trong đó M là hằng số không phụ thuộc h.

Vậy công thức (6.12) chính xác hơn công thức Euler. Tuy nhiên nó có nhược điểm là y_{i+1} xuất hiện cả ở vế phải. Như vậy khi đã biết y_i ta vẫn còn phải giải một phương trình đại số phi tuyến đối với y_{i+1} (nếu f(x,y) phi tuyến đối với y). Vì vậy đây là một phương pháp ẩn. Người ta đã cải tiến phương pháp (6.12) bằng cách phối hợp (6.12) với phương pháp Euler như sau:

$y_0 = y(x_0)$ đã biết

Với i =1, 2,... n ta tính

$$z = y_{i-1} + hf(x_{i-1},y_{i-1})$$

$$y_i = y_{i-1} + \frac{h}{2}\,[f(x_{i-1},y_{i-1}) + f(x_i,z)] \qquad (6.13)$$

i=0,1,...,n-1, m=1,2,...

Công thức (6.13) được gọi là công thức Euler cải tiến. Như vậy trong (6.13) đầu tiên người ta dùng công thức Euler để ước lượng giá trị của y_i (được ký hiệu là z) và dùng z để tính

$$y_i = y_{i-1} + \frac{h}{2}\,[f(x_{i-1},y_{i-1}) + f(x_i,z)]$$

Ta có thể mô tả thuật toán ***Euler cải tiến*** để cài đặt trên máy tính theo các bước sau:

a.Thuật toán cho một lần chia khoảng duy nhất.

Nhập a, b , y_0 và n.

Đặt $h = \frac{b-a}{n}$, $x_0 = a$.

Với i =1,2,..., n tính

$$z = y_{i-1} + hf(x_{i-1},y_{i-1})$$

$$y_i = y_{i-1} + \frac{h}{2}\,[f(x_{i-1},y_{i-1}) + f(x_i,z)]$$

$$x_i = x_{i-1} + h$$

b.Thuật toán cho nhiều lần chia khoảng.

- **Bước 0:** Nhập a, b , y_0 , kmax và $\varepsilon>0$.

Đặt $h_0 = \frac{b-a}{n}$, $x_0^{(0)} = a$, $y_0^{(0)} = y_0$

Với i = 1, 2,... tính

$$x_i^{(0)} = x_{i-1}^{(0)} + h_0$$

$$z = y_{i-1}^{(0)} + h_0 f(x_{i-1}^{(0)}, y_{i-1}^{(0)})$$

$$y_i^{(0)} = y_{i-1}^{(0)} + \frac{h_0}{2}[f(x_{i-1}^{(0)}, y_{i-1}^{(0)}) + f(x_i^{(0)}, z)]$$

- **Bước 1:** Đặt $h_1 = \frac{h_0}{2}$, $x_0^{(1)} = a$, $y_0^{(1)} = y_0$

Với i = 1, 2,... tính

$$x_i^{(1)} = x_{i-1}^{(1)} + h_1$$

$$z = y_{i-1}^{(1)} + h_1 f(x_{i-1}^{(1)}, y_{i-1}^{(1)})$$

$$y_i^{(1)} = y_{i-1}^{(1)} + \frac{h_1}{2}[f(x_{i-1}^{(1)}, y_{i-1}^{(1)}) + f(x_i^{(1)}, z)]$$

$$d_1 = \max_i |y_{2i}^{(1)} - y_i^{(0)}|, i=0,1,2,...,n$$

Nếu $d_1 < \varepsilon$ thì dừng thuật toán và lấy mẫu (x_0,y_0), (x_1,y_1), ..., (x_{2n},y_{2n}) làm nghiệm xấp xỉ.

Nếu 2 điều trên đây không xẩy ra thì chuyển qua bước (3).

...

- **Bước k:** Đặt $h_k = \frac{h_{k-1}}{2}$, $x_0^{(k)} = a$, $y_0^{(k)} = y_0$

Với i = 1, 2,... tính

$$x_i^{(k)} = x_{i-1}^{(k)} + h_k$$

$$z = y_{i-1}^{(k)} + h_k f(x_{i-1}^{(k)}, y_{i-1}^{(k)})$$

$$y_i^{(k)} = y_{i-1}^{(k)} + \frac{h_k}{2}[f(x_{i-1}^{(k)}, y_{i-1}^{(k)}) + f(x_i^{(k)}, z)]$$

$$d_k = \max_i |y_{2i}^{(k)} - y_i^{(k-1)}|, i=0,1,2,...,n.2^{k-1}$$

Nếu $d_k < \varepsilon$ thì dừng thuật toán và lấy mẫu (x_0,y_0), (x_1,y_1), ..., (x_N,y_N), trong đó $N = n.2^k$ làm nghiệm xấp xỉ.

Nếu $k \geq kmax$ thì thông báo phép lặp chưa hội tụ và cũng dừng thuật toán.

Nếu 2 điều trên đây không xẩy ra thì chuyển qua bước (k+1).

6.4. PHƯƠNG PHÁP EULER-CAUCHY

Thực chất phương pháp Euler-Cauchy cũng là phương pháp Euler cải tiến. Tuy nhiên sự khác biệt là khi tính y_i ta dùng công thức (6.13) nhiều lần để đạt được độ chính xác cao hơn. Cụ thể từ $h = \frac{b-a}{n}$, $x_0 = a$ ta tính các $x_i = x_{i-1} + h$.

Từ điều kiện ban đầu $y_0 = y(x_0)$, cho trước $\delta > 0$, với i = 1, 2, ... ta thực hiện thuật toán sau:

- **Bước 1:**

(a) $u = y_0 + hf(x_0,y_0)$

(b) $v = y_0 + \frac{h}{2}\,[f(x_0,y_0) + f(x_0,u)]$ (6.14)

Nếu $|v-u| < \delta$ thì ta chọn $y_1 = v$, ngược lại ta đặt u = v và tính lại (b).

Người ta chứng minh được rằng với h đủ bé thì quá trình lặp (6.14) hội tụ. Vì vậy nếu sau ba bốn lần lặp mà vẫn không đạt được sự trùng nhau đến mức đòi hỏi của các gần đúng liên tiếp thì cần giảm bước h và làm lại từ đầu.

. . .

- **Bước i:**

(a) $u = y_{i-1} + hf(x_{i-1},y_{i-1})$

(b) $v = y_{i-1} + \frac{h}{2}\,[f(x_{i-1},y_{i-1}) + f(x_i,u)]$ (6.15)

Nếu $|v-u| < \delta$ thì ta chọn $y_i = v$, ngược lại ta đặt u = v và tính lại (b).

Người ta chứng minh được rằng với h đủ bé thì quá trình lặp (6.15) hội tụ. Vì vậy nếu sau ba bốn lần lặp mà vẫn không đạt được sự trùng nhau đến mức đòi hỏi của các gần đúng liên tiếp thì cần giảm bước h và làm lại từ đầu.

Ta có thể mô tả thuật toán ***Euler - Cauchy*** để cài đặt trên máy tính theo các bước sau:

a. Thuật toán cho một lần chia khoảng duy nhất.

Nhập a, b , y_0, δ và n.

Đặt $h = \frac{b-a}{n}$, $x_0 = a$ ta tính các $x_i = x_{i-1} + h$.

Từ điều kiện ban đầu $y_0 = y(x_0)$, với i = 1, 2, ... ta thực hiện thuật toán như (6.14) và (6.15).

b. Thuật toán cho nhiều lần chia khoảng.

- **Bước 0:** Nhập a, b , y_0 , δ, n, kmax và ε.

Đặt $h_0 = \frac{b-a}{n}$, $x_0^{(0)} = a$, $y_0^{(0)} = y_0$

Với i = 1, 2,... tính các $y_i^{(0)}$ theo (6.14) và (6.15).

- **Bước 1:** Đặt $h_1 = \frac{h_0}{2}$, $x_0^{(1)} = a$, $y_0^{(1)} = y_0$

Với i = 1, 2,... tính các $y_i^{(1)}$ theo (6.14) và (6.15).

Tính $d_1 = \max_i |y_{2i}^{(1)} - y_i^{(0)}|$, i=0,1,2,...,n

Nếu $d_1 < \varepsilon$ thì dừng thuật toán và lấy mẫu $(x_0,y_0), (x_1,y_1), ..., (x_{2n},y_{2n})$ làm nghiệm xấp xỉ.

Nếu 2 điều trên đây không xẩy ra thì chuyển qua bước (3).

...

- **Bước k:** Đặt $h_k = \frac{h_{k-1}}{2}$, $x_0^{(k)} = a$, $y_0^{(k)} = y_0$

Với i = 1, 2,... tính các $y_i^{(k)}$ theo (6.14) và (6.15).

Tính $d_k = \max_i | y_{2i}^{(k)} - y_i^{(k-1)} |$, i=0,1,2,...,$n.2^{k-1}$

Nếu $d_k < \varepsilon$ thì dừng thuật toán và lấy mẫu $(x_0,y_0), (x_1,y_1), ..., (x_N,y_N)$, trong đó $N = n.2^k$ làm nghiệm xấp xỉ.

Nếu $k \geq kmax$ thì thông báo phép lặp chưa hội tụ và cũng dừng thuật toán.

Nếu 2 điều trên đây không xẩy ra thì chuyển qua bước (k+1).

6.5. PHƯƠNG PHÁP RUNGE - KUTTA

Phương pháp Runge - Kutta là phương pháp rất hiệu quả: nó vừa có độ chính xác cao lại vừa là phương pháp hiện. Để thành lập những công thức Runge-Kutta có độ chính xác cao hơn công thức Euler, người ta dùng khai triển Taylo nghiệm y(x) tại x_i với nhiều số hạng hơn. Xây dựng công thức Runge-Kutta trong trường hợp tổng quát khá phức tạp, ở đây ta chỉ xét trường hợp đơn giản nhất.

Trở lại xét bài toán (6.1), ta xét khai triển Taylor của nghiệm đúng y(x) :

$$y(x) = y(x_i) + \frac{x-x_i}{1!}y'(x_i) + \frac{(x-x_i)^2}{2!}y''(x_i) + \frac{(x-x_i)^3}{3!}y'''(c_i) , \ c_i \in (x_i,x)$$

Thay $x = x_{i+1} = x_i + h$, ta có

$$y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{6}y'''(c_i), \ c \in (x_i,x) \qquad (6.16)$$

Trong đó

$$y'(x_i) = f(x_i,y(x_i))$$

$$y''(x_i) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} \ | \ x=x_i$$

$$= f_x'(x_i,y_i) + f_y'(x_i,y(x_i))y'(x_i)$$

Thay vào (6.13) ta có

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2}[f_x'(x_i,y_i) + f_y'(x_i,y_i)y'_i] + O(h^3) \qquad (6.17)$$

Để tránh tính trực tiếp $f_x'(x_i,y_i)$ và $f_y'(x_i,y_i)$, Runge và Kutta đã làm như sau:

Đặt

$$y_{i+1} = y_i + r_1k_1^{(i)} + r_2k_2^{(i)} \qquad (6.18)$$

Trong đó

$$k_1^{(i)} = hf(x_i, y_i)$$

$$k_2^{(i)} = hf(x_i + \alpha h,\ y_i + \beta k_1^{(i)}) \qquad (6.19)$$

và chọn α, β, r_1, r_2 sao cho khai triển theo lũy thừa của h của y_{i+1} xác định bởi (6.18) trùng nhau đến 3 số hạng đầu của vế phải công thức (6.17).

Dùng công thức Taylor của hàm hai biến, ta có:

$$f(x_i + \alpha h,\ y_i + \beta k_1^{(i)}) = f(x_i,y_i) + \alpha h f_x'(x_i,y_i) + \beta k_1^{(i)} f_y'(x_i,y_i) + O(h^2) =$$

$$= y_i' + \alpha h f_x'(x_i,y_i) + \beta k_1^{(i)} f_y'(x_i,y_i) + O(h^2)$$

Từ đây ta có

$$k_1^{(i)} = hf(x_i, y_i) = hy_i'$$

$$k_2^{(i)} = hf(x_i + \alpha h,\ y_i + \beta k_1^{(i)}) =$$

$$= h\, y_i' + \alpha h^2 f_x'(x_i,y_i) + \beta h^2 y_i' f_y'(x_i,y_i) + O(h^3)$$

Do đó (6.18) có thể viết dưới dạng

$$y_{i+1} = y_i + r_1 h\, y_i' + r_2[\, h\, y_i' + \alpha h^2 f_x'(x_i,y_i) + \beta h^2 y_i' f_y'(x_i,y_i)] + O(h^3) =$$

$$= y_i + r_1 h\, y_i' + r_2 h\, y_i' + \alpha\, r_2 h^2 f_x'(x_i,y_i) + \beta\, r_2\, h^2 y_i' f_y'(x_i,y_i) + O(h^3) =$$

$$= y_i + (r_1 + r_2)\, h\, y_i' + h^2 (\alpha r_2 f_x'(x_i,y_i) + \beta\, r_2\, f_y'(x_i,y_i)\, y_i'] + O(h^3) \quad (6.20)$$

So sánh các hệ số lũy thừa của h trong (6.17) và (6.20) ta có

$$r_1 + r_2 = 1$$

$$\alpha r_2 = \beta\, r_2 = 1/2$$

Đây là một hệ thống 3 phương trình, 4 ẩn số nên là một hệ vô định. Ta xét một vài họ nghiệm đơn giản

(1) $r_1 = 0$, $r_2 = 1$, $\alpha = \beta = 1/2$. Khi đó (6.18) và (6.19) có dạng

$$y_0 = y(x_0) \text{ đã biết}$$

$$k_1^{(i)} = hf(x_i, y_i)$$

$$k_2^{(i)} = hf(x_i + h/2,\ y_i + k_1^{(i)}/2)$$

$$y_{i+1} = y_i + k_2^{(i)} \quad i=0,1,...,n-1 \qquad (6.21)$$

(2) $r_1 = r_2 = 1/2$, $\alpha = \beta = 1$. Khi đó (6.18) và (6.19) có dạng

$$y_0 = y(x_0) \text{ đã biết}$$

$$k_1^{(i)} = hf(x_i, y_i)$$

$$k_2^{(i)} = hf(x_i + h,\ y_i + k_1^{(i)})$$

$$y_{i+1} = y_i + (1/2)(\, k_1^{(i)} + k_2^{(i)}) \qquad i=0,1,...,n-1 \qquad (6.22)$$

Khi thành lập các công thức (6.18) và (6.19) trên đây ta bỏ qua số hạng $O(h^3)$ trong khai triển Taylor. Ta có thể chứng minh được rằng sai số tại điểm x_i thỏa mãn:

$|y_i - y(x_i)| \le Mh^2$, trong đó M là hằng số dương không phụ thuộc h.

Vậy các phương pháp Runge-Kutta trên đây có độ chính xác cấp hai.

Hoàn toàn tương tự, nếu trong khai triển Taylor của $y(x_{i+1})$ tại x_i ta bỏ qua số hạng $o(h^4)$ thì sẽ nhận được công thức Runge-Kutta có độ chính xác cấp ba, nghĩa là

$|y_i - y(x_i)| \le Mh^3$, trong đó M là hằng số dương không phụ thuộc h.

$y_0 = y(x_0)$ đã biết

$k_1^{(i)} = hf(x_i, y_i)$

$k_2^{(i)} = hf(x_i + h/2,\ y_i + k_1^{(i)}/2)$

$k_3^{(i)} = hf(x_i + h,\ y_i - k_1^{(i)} + 2k_2^{(i)})$

$$y_{i+1} = y_i + (1/6)(k_1^{(i)} + 4k_2^{(i)} + k_3^{(i)}) \quad i=0,1,...,n-1 \qquad (6.23)$$

Nếu bỏ qua số hạng $O(h^5)$ thì ta nhận được công thức Runge-Kutta có độ chính xác cấp 4:

$y_0 = y(x_0)$ đã biết

$k_1^{(i)} = hf(x_i, y_i)$

$k_2^{(i)} = hf(x_i + h/2,\ y_i + k_1^{(i)}/2)$

$k_3^{(i)} = hf(x_i + h/2,\ y_i + k_2^{(i)}/2)$

$k_4^{(i)} = hf(x_i + h/2,\ y_i + k_3^{(i)})$

$$y_{i+1} = y_i + (1/6)(k_1^{(i)} + 2k_2^{(i)} + 2k_3^{(i)} + k_4^{(i)}) \quad i=0,1,...,n-1 \qquad (6.24)$$

Trong các công thức Runge-Kutta nêu trên người ta thường dùng công thức (6.24) vì nó có độ chính xác cao mà lại không quá phức tạp. Trong thực tế việc xác định hằng số M trong đánh giá sai số của phương pháp Runge-Kutta khá phức tạp, do đó người ta thường xác định sai số bằng cách "tính 2 lần" như sau:

Lần đầu tính bằng công thức (6.24) với bước h, nhận được $y_n^{(h)}$ là giá trị gần đúng của y(b) . Sau đó ta lại tính với bước h/2 nhận được $y_{2n}^{(\frac{h}{2})}$ là giá trị gần đúng của y(b) và sai số được xác định bởi:

$$| y_{2n}^{(\frac{h}{2})} - y(b) | \approx (1/15) | y_{2n}^{(\frac{h}{2})} - y_n^{(h)} | \qquad (6.25)$$

Ví dụ.

Cho bài toán Cauchy như sau:

$y' = x + y,\ y(0) = 1$

Hãy tìm nghiệm gần đúng bằng phương pháp Runge-Kutta (6.24) trên [0,0.5] với bước h=0.1

Giải: Ta có $x_i = 0.1i$; i = 0,1,2,3,4,5

$y_0 = 1$

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

$k_2^{(0)} = 0.1(0+0.05) + (1+0.05)] = 0.11$

$k_3^{(0)} = 0.1(0+0.05) + (1 + 0.055)] = 0.1105$

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

Từ đó

$$y_1 = 1 + \frac{1}{6}(0.1 + 2*0.11 + 2*0.1105 + 0.12105) = 1.1103$$

Tương tự ta có thể tính y_2, y_3, y_4 và y_5.

Ta có thể thấy rằng $y(0.5) \approx y_5 = 1.7974$

Nghiệm đúng của bài toán Cauchy đã cho là $y = 2e^x - x - 1$

từ đó:

$$y(0.5) = 2\, e^{0.5} - 0.5 - 1 = 1.79744$$

Như vậy kết quả nhận được dúng đến 4 số lẻ thập phân.

Các phương pháp Runge - Kutta nêu trên đều có thể áp dụng cho hệ phương trình vi phân cấp một.

Bạn đọc có thể tự viết chương trình theo công thức (6.24) và thử in ra các kết quả trên đây.

6.5. BÀI TẬP

Bài 1. Giải phương trình sau bằng phương pháp Euler

$y' = \frac{xy}{2}$; $x \in [0,1]$, $y(0) = 1$; $h = 0,1$

Bài 2. Giải phương trình sau bằng phương pháp Euler

$y' = x^2 + y^2$; $x \in [0,1]$, $y(0) = 1$; $h = 0,2$

Bài 3. Giải phương trình sau bằng phương pháp Runge-Kutta:

$y' = y - 2x/y$; $x \in [0,1]$, $y(0) = 1$; $h = 0,2$

Bài 4. Giải bài toán sau bằng phương pháp Euler cải tiến và so sánh kết quả với nghiệm đúng:

$y' = x^2 + y^2$; $x \in [0,1]$, $y(0) = 1$; $h = 0,2$

Bài 5. Thử lại hoặc viết mới các chương trình cài đặt các thuật toán Euler rồi chạy thử để kiểm tra các kết quả trên đây.

TÓM TẮT NỘI DUNG CHƯƠNG 6

Trong chương này chúng ta cần chú ý nhất là các vấn đề sau:

1. Bài toán Cauchy

Tìm hàm y=y(x) thỏa mãn:

$y'(x) = f(x,y) \quad x \in [a,b], \; x_0 = a$ (6.1)

$y(x_0) = y_0$ (6.1b)

Điều kiện (6.1b) được gọi là điều kiện ban đầu hay điều kiện Cauchy.

2. Một số phương pháp tìm nghiệm gần đúng của bài toán Cauchy

a.Phương pháp EULER

b.Phương pháp EULER cải tiến

c. Phương pháp EULER-CAUCHY

d.Phương pháp RUNGE-KUTTA