Chapter 6 Initial-Value Problems for Ordinary Differential Equations

6-1 Euler's Method

Solve $y'=f(t,y)$, $y(a)=\alpha$, $a \leqq t \leqq b$. Set $h=(b-a)/n$, $t_i=a+ih$, $i=0, 1, 2, \ldots, n$

$$\frac{y(t_{i+1})-y(t_i)}{h}=f(t_i, y(t_i)) \Rightarrow y(t_{i+1})=y(t_i)+h \cdot f(t_i,y(t_i))$$

Eg. Solve $y'=-y+t+1$, $0 \leqq t \leqq 1$, $y(0)=1$.

(Sol.) Set $h=\frac{1-0}{10}=0.1$, $t_i=0.1i$, $y(t_{i+1})=y(t_i)+h \cdot [-y(t_i)+t_i+1]$

C++ Program:

```
#include <stdio.h>
#include <math.h>
main()
{float a,b,h,y0,t; int i,n;
printf("a,b,n,y(a)\n");
scanf("%f %f %d %f",&a,&b,&n,&y0);
h=(b-a)/n;
float y=y0;
for (i=1;i<=n;i++)
  {
   t=a+i*h;
   y=y+h*(-y+t+1);
   printf("%f %f\n",t,y);
  }
}
```

```
a,b,n,y(a)
0 1 10 1
0.100000 1.010000
0.200000 1.029000
0.300000 1.056100
0.400000 1.090490
0.500000 1.131441
0.600000 1.178297
0.700000 1.230467
0.800000 1.287420
0.900000 1.348678
1.000000 1.413811
Press any key to continue
```

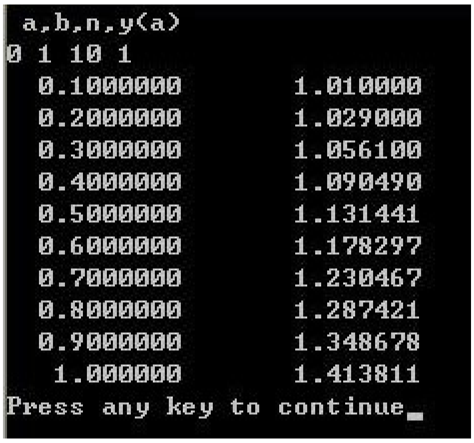
Fortran Program 1:

```
      real h
      write (*,*) 'a,b,n,y(a)'
      read (*,*) a,b,n,y0
      h=(b-a)/n
      y=y0
      do 1 i=1,n
        t=a+i*h
        y=y+h*(-y+t+1)
        write (*,*) t,y
1     continue
      stop
      end
```

```
 a,b,n,y(a)
0 1 10 1
  0.1000000        1.010000
  0.2000000        1.029000
  0.3000000        1.056100
  0.4000000        1.090490
  0.5000000        1.131441
  0.6000000        1.178297
  0.7000000        1.230467
  0.8000000        1.287421
  0.9000000        1.348678
   1.000000        1.413811
Press any key to continue
```

Fortran Program 2:

```
      real h
      write (*,*) 'a,b,n,y(a)'
      read (*,*) a,b,n,y0
      h=(b-a)/n
      y=y0
      do 1 i=1,n
        t=a+i*h
        y=y+h*f(t,y)
        write (*,*) t,y
1     continue
      stop
      end

      function f(t,y)
        f=-y+t+1
        return
      end
```



```
 a,b,n,y(a)
0 1 10 1
  0.1000000        1.010000
  0.2000000        1.029000
  0.3000000        1.056100
  0.4000000        1.090490
  0.5000000        1.131441
  0.6000000        1.178297
  0.7000000        1.230467
  0.8000000        1.287421
  0.9000000        1.348678
   1.000000        1.413811
Press any key to continue_
```

Solve $\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_m(t) \end{bmatrix} = \begin{bmatrix} f_1(t,x_1,x_2,\cdots,x_m) \\ f_2(t,x_1,x_2,\cdots,x_m) \\ \vdots \\ f_m(t,x_1,x_2,\cdots,x_m) \end{bmatrix}$, $\begin{bmatrix} x_1(a) \\ x_2(a) \\ \vdots \\ x_m(a) \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}$, $a \leqq t \leqq b$.

Set $h=(b-a)/n$, $t_i=a+ih$, $i=0, 1, 2, \ldots, n$

$$\Rightarrow \begin{bmatrix} x_1(t_{i+1}) \\ x_2(t_{i+1}) \\ \vdots \\ x_m(t_{i+1}) \end{bmatrix} = \begin{bmatrix} x_1(t_i)+h\cdot f_1(t_i,x_1(t_i),x_2(t_i),\cdots,x_m(t_i)) \\ x_2(t_i)+h\cdot f_2(t_i,x_1(t_i),x_2(t_i),\cdots,x_m(t_i)) \\ \vdots \\ x_m(t_i)+h\cdot f_m(t_i,x_1(t_i),x_2(t_i),\cdots,x_m(t_i)) \end{bmatrix}$$

Eg. Solve $y''-2y'-3y=t$, $y(0)=1$, $y'(0)=-1/3$.

(Sol.) Let $x_1=y$, $x_2=y'=x_1'$, then we have $\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ t+3x_1(t)+2x_2(t) \end{bmatrix}$

Set $h=(1-0)/20=0.05$, $t_i=0.05i$, $i=0, 1, \ldots, 20 \Rightarrow \begin{bmatrix} x_1(t_{i+1}) \\ x_2(t_{i+1}) \end{bmatrix} = \begin{bmatrix} x_1(t_i)+h\cdot x_2(t_i) \\ x_2(t_i)+h\cdot[t_i+3x_1(t_i)+2x_2(t_i)] \end{bmatrix}$,

where $x_1(t_0)=x_1(0)=1$, $x_2(t_0)=x_2(0)=-1/3$

$$\Rightarrow \begin{bmatrix} x_1(t_1) \\ x_2(t_1) \end{bmatrix} = \begin{bmatrix} 1+h\cdot x_2(t_0) \\ -1/3+h\cdot[t_0+3x_1(t_0)+2x_2(t_0)] \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x_1(t_2) \\ x_2(t_2) \end{bmatrix} = \begin{bmatrix} x_1(t_1)+h\cdot x_2(t_1) \\ x_2(t_1)+h\cdot[t_1+3x_1(t_1)+2x_2(t_1)] \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x_1(t_3) \\ x_2(t_3) \end{bmatrix} = \begin{bmatrix} x_1(t_2)+h\cdot x_2(t_2) \\ x_2(t_2)+h\cdot[t_2+3x_1(t_2)+2x_2(t_2)] \end{bmatrix}, \cdots$$

Fortran Program:

```
        real h
        write (*,*) 'a,b,n,y(a),dy(a)'
        read (*,*) a,b,n,x10,x20
        h=(b-a)/n
        x1=x10
        x2=x20
        do 1 i=1,n
          t=a+i*h
          x1=x1+h*x2
          x2=x2+h*(t+3*x1+2*x2)
          write (*,*) t,x1,x2
1       continue
        stop
        end
```

```
 a,b,n,y(a),dy(a)
0 1 20 1 -1/3
  5.0000001E-02  0.9500000     -0.9550000
  0.1000000      0.9022500     -0.9101625
  0.1500000      0.8567418     -0.8651675
  0.2000000      0.8134835     -0.8196617
  0.2500000      0.7725004     -0.7732528
  0.3000000      0.7338377     -0.7255025
  0.3500000      0.6975626     -0.6759183
  0.4000000      0.6637667     -0.6239452
  0.4500000      0.6325694     -0.5689543
  0.5000000      0.6041217     -0.5102314
  0.5500000      0.5786102     -0.4469630
  0.6000000      0.5562620     -0.3782201
  0.6500000      0.5373510     -0.3029394
  0.7000000      0.5222040     -0.2199028
  0.7500000      0.5112089     -0.1277117
  0.8000000      0.5048233     -2.4759367E-02
  0.8500000      0.5035853      9.0802498E-02
  0.9000000      0.5081255      0.2211016
  0.9500000      0.5191805      0.3685888
   1.000000      0.5376100      0.5360892
Press any key to continue
```

Note: Euler's method leads the accumulation of numerical errors.

6-2 The forth-order Runge-Kutta Method

Solve $y'=f(t,y)$, $a \leqq t \leqq b$, $y(a)=\alpha$. Set $h=(b-a)/n$, $t_i=a+ih$, $i=0, 1, 2, \ldots, n$

$$\begin{cases} K_1 = h\cdot f(t_i, y(t_i)) \\ K_2 = h\cdot f\left(t_i+\frac{h}{2}, y(t_i)+\frac{K_1}{2}\right) \\ K_3 = h\cdot f\left(t_i+\frac{h}{2}, y(t_i)+\frac{K_2}{2}\right) \\ K_4 = h\cdot f(t_i+h, y(t_i)+K_3) \end{cases} \Rightarrow y(t_{i+1}) \approx y(t_i)+\frac{1}{6}(K_1+2K_2+2K_3+K_4)\ ,\ \text{where}$$

$i=0, 1, 2, \ldots, n$

Solve $\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_m(t) \end{bmatrix} = \begin{bmatrix} f_1(t,x_1(t),\cdots,x_m(t)) \\ f_2(t,x_1(t),\cdots,x_m(t)) \\ \vdots \\ f_m(t,x_1(t),\cdots,x_m(t)) \end{bmatrix}$, $\begin{matrix} x_1(a)=\alpha_1 \\ x_2(a)=\alpha_2 \\ \vdots \quad \vdots \\ x_m(a)=\alpha_m \end{matrix}$, $a \leqq t \leqq b$

$$\begin{bmatrix} K_{11} \\ \vdots \\ K_{1m} \end{bmatrix} = h\cdot \begin{bmatrix} f_1(t_i,x_1(t_i),\cdots,x_m(t_i)) \\ \vdots \quad \vdots \quad \vdots \\ f_m(t_i,x_1(t_i),\cdots,x_m(t_i)) \end{bmatrix}$$

$$\begin{bmatrix} K_{21} \\ \vdots \\ K_{2m} \end{bmatrix} = h\cdot \begin{bmatrix} f_1\left(t_i+\frac{h}{2}, x_1(t_i)+\frac{K_{11}}{2},\cdots,x_m(t_i)+\frac{K_{1m}}{2}\right) \\ \vdots \quad \vdots \quad \vdots \\ f_m\left(t_i+\frac{h}{2}, x_1(t_i)+\frac{K_{11}}{2},\cdots,x_m(t_i)+\frac{K_{1m}}{2}\right) \end{bmatrix}$$

$$\begin{bmatrix} K_{31} \\ \vdots \\ K_{3m} \end{bmatrix} = h\cdot \begin{bmatrix} f_1\left(t_i+\frac{h}{2}, x_1(t_i)+\frac{K_{21}}{2},\cdots,x_m(t_i)+\frac{K_{2m}}{2}\right) \\ \vdots \quad \vdots \quad \vdots \\ f_m\left[t_i+\frac{h}{2}, x_1(t_i)+\frac{K_{21}}{2},\cdots,x_m(t_i)+\frac{K_{2m}}{2}\right] \end{bmatrix}$$

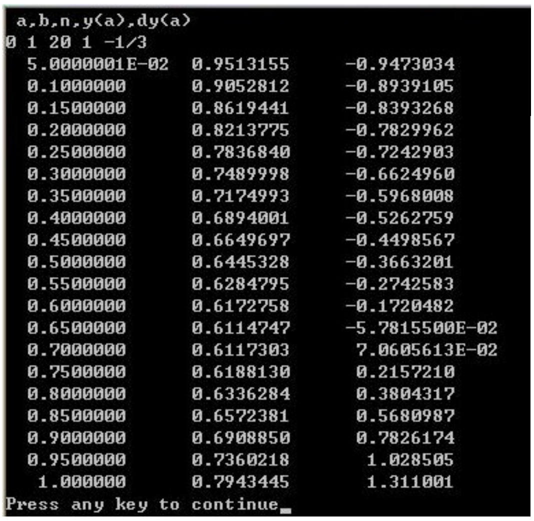
$$\begin{bmatrix} K_{41} \\ \vdots \\ K_{4m} \end{bmatrix} = h\cdot \begin{bmatrix} f_1(t_i+h, x_1(t_i)+K_{31},\cdots,x_m(t_i)+K_{3m}) \\ \vdots \quad \vdots \quad \vdots \\ f_m(t_i+h, x_1(t_i)+K_{31},\cdots,x_m(t_i)+K_{3m}) \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} x_1(t_{i+1}) \\ \vdots \\ x_m(t_{i+1}) \end{bmatrix} = \begin{bmatrix} x_1(t_i) \\ \vdots \\ x_m(t_i) \end{bmatrix} + \frac{1}{6}\begin{bmatrix} K_{11}+2K_{21}+2K_{31}+K_{41} \\ \vdots \\ K_{1m}+2K_{2m}+2K_{3m}+K_{4m} \end{bmatrix}$$

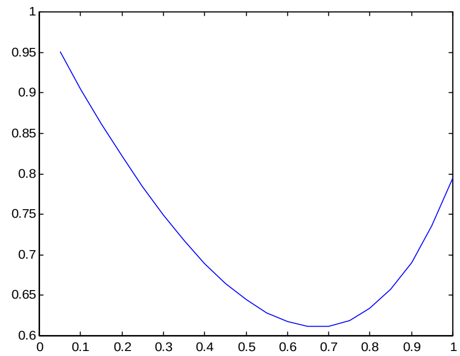
Eg. Solve $y''-2y'-3y=t$, $y(0)=1$, $y'(0)=-1/3$, by the Runge-Kutta method.

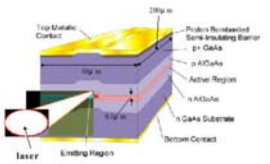
```
real h,k11,k12,k21,k22,k31,k32,k41,k42
write (*,*) 'a,b,n,y(a),dy(a)'
read (*,*) a,b,n,x10,x20
```

```
        h=(b-a)/n
        x1=x10
        x2=x20
        do 1 i=1,n
         t=a+i*h
         k11=h*f1(t,x1,x2)
         k12=h*f2(t,x1,x2)
         k21=h*f1(t+h/2,x1+k11/2,x2+k12/2)
         k22=h*f2(t+h/2,x1+k11/2,x2+k12/2)
         k31=h*f1(t+h/2,x1+k21/2,x2+k22/2)
         k32=h*f2(t+h/2,x1+k21/2,x2+k22/2)
         k41=h*f1(t+h,x1+k31,x2+k32)
         k42=h*f2(t+h,x1+k31,x2+k32)
         x1=x1+(k11+2*k21+2*k31+k41)/6.
         x2=x2+(k12+2*k22+2*k32+k42)/6.
         write (*,*) t,x1,x2
1       continue
        stop
        end

        function f1(t,x1,x2)
         f1=x2
         return
        end

        function f2(t,x1,x2)
         f2=t+3*x1+2*x2
         return
        end
```



```
 a,b,n,y(a),dy(a)
0 1 20 1 -1/3
  5.0000001E-02  0.9513155     -0.9473034
  0.1000000      0.9052812     -0.8939105
  0.1500000      0.8619441     -0.8393268
  0.2000000      0.8213775     -0.7829962
  0.2500000      0.7836840     -0.7242903
  0.3000000      0.7489998     -0.6624960
  0.3500000      0.7174993     -0.5968008
  0.4000000      0.6894001     -0.5262759
  0.4500000      0.6649697     -0.4498567
  0.5000000      0.6445328     -0.3663201
  0.5500000      0.6284795     -0.2742583
  0.6000000      0.6172758     -0.1720482
  0.6500000      0.6114747     -5.7815500E-02
  0.7000000      0.6117303      7.0605613E-02
  0.7500000      0.6188130      0.2157210
  0.8000000      0.6336284      0.3804317
  0.8500000      0.6572381      0.5680987
  0.9000000      0.6908850      0.7826174
  0.9500000      0.7360218       1.028505
   1.000000      0.7943445       1.311001
Press any key to continue_
```

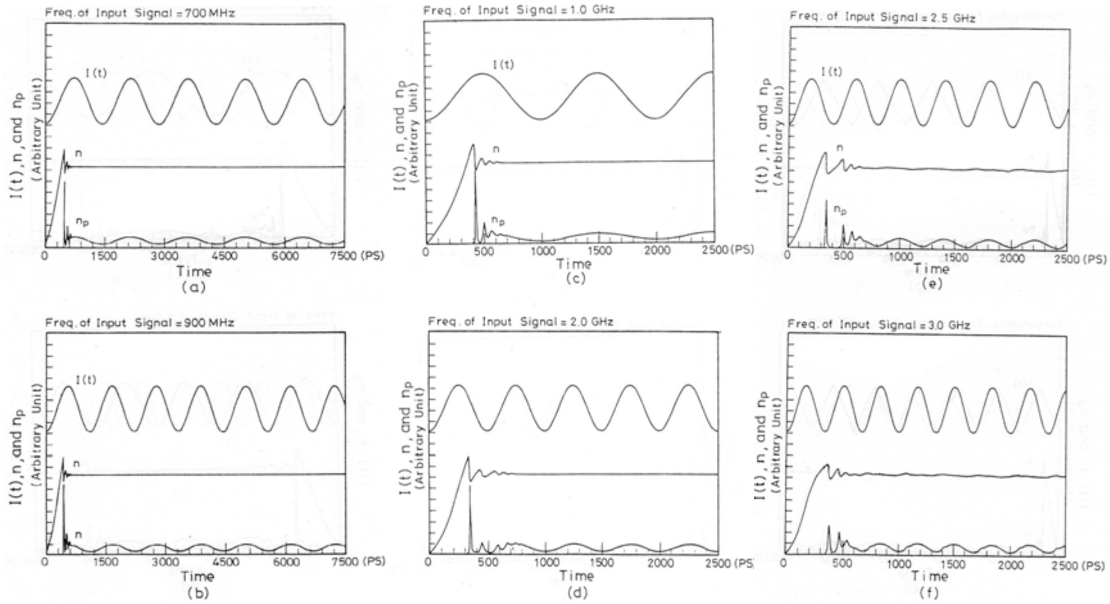




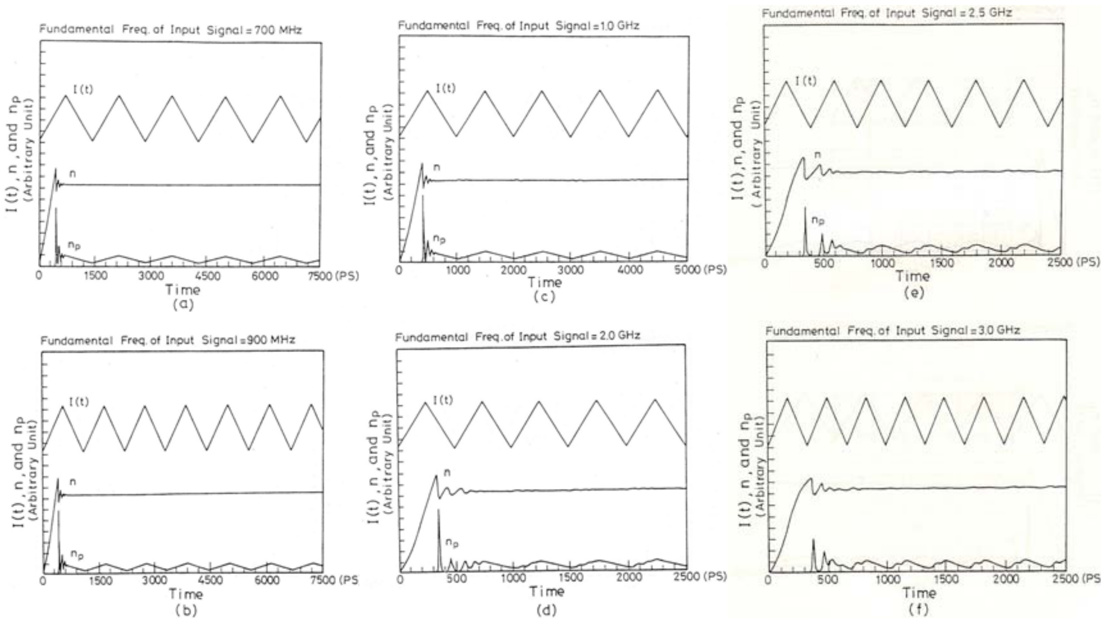
Eg. Solve the rate equations for describing the characteristics of the laser diode:

$$\begin{cases} dn/dt = I(t)/qV - Bn^2 - g(t)v_g n_p - n/\tau \\ dn_p/dt = g(n)v_g n_p - n_p/\tau_p + \beta B n^2 \end{cases}.$$

(Sol.) For the sinusoidal input current $I(t)$:



For the triangular input current $I(t)$:



6-3 Multi-step Methods

The 4th-order Adams-Bashforth method: Solve $y'=f(t,y)$ by

$$y(t_{i+1})=y(t_i)+\frac{h}{24}[55f(t_i,y(t_i))-59f(t_{i-1},y(t_{i-1}))+37f(t_{i-2},y(t_{i-2}))-9f(t_{i-3},y(t_{i-3}))],$$

where $h=(b-a)/n$, $t_i=a+ih$, $i=0, 1, 2, \ldots, n$, and $y(t_0)$, $y(t_1)$, $y(t_2)$, and $y(t_3)$ are given.

The 4th-order Adams-Moulton method: Solve $y'=f(t,y)$ by

$$y(t_{i+1})=y(t_i)+\frac{h}{24}[9f(t_{i+1},y(t_{i+1}))+19f(t_i,y(t_i))-5f(t_{i-1},y(t_{i-1}))+f(t_{i-2},y(t_{i-2}))],$$

where $y(t_0)$, $y(t_1)$, and $y(t_2)$ are given.

Note: Adams-Bashforth technique is an **explicit** four-step method.

Adams-Moulton technique is an **implicit** four-step method.

Eg. Solve $y'=-y+t+1$, $0\leqq t\leqq 1$, $y(0)=1$.

(Sol.) $f(t,y)=-y+t+1$, set $h=\frac{1-0}{10}=0.1$, $t_i=0+0.1i=0.1i$

Adams-Bashforth method:

$$y_{i+1}=\frac{1}{24}[18.5y_i+5.9y_{i-1}-3.7y_{i-2}+0.9y_{i-3}+0.24i+2.52]$$

Adams-Moulton method:

$$y_{i+1}=\frac{1}{24}[-0.9y_{i+1}+22.1y_i+0.5y_{i-1}-0.1y_{i-2}+0.24i+2.52]$$

$$\Rightarrow y_{i+1}=\frac{1}{24.9}[22.1y_i+0.5y_{i-1}-0.1y_{i-2}+0.24i+2.52]$$

Predictor-corrector method: Combination of explicit and implicit techniques as

$$\begin{cases} y_{i+1}^{(0)}=y_i+\frac{h}{24}[55f(t_i,y_i)-59f(t_{i-1},y_{i-1})+37f(t_{i-2},y_{i-2})-9f(t_{i-3},y_{i-3})] \\ y_{i+1}^{(1)}=y_i+\frac{h}{24}[9f(t_{i+1},y_{i+1}^{(0)})+19f(t_i,y_i)-5f(t_{i-1},y_{i-1})+f(t_{i-2},y_{i-2})], \quad a\le t\le b \end{cases},$$

where $h=(b-a)/n$, $y_i=y(t_i)$, $t_i=a+ih$, $i=0, 1, 2, \ldots, n$.

Root condition **Let λ_1, λ_2, …, λ_m (not necessarily distinct) be the roots of the characteristic polynomial equation $p(\lambda)=\lambda^m-a_{m-1}\lambda^{m-1}-\ldots-a_1\lambda-a_0=0$, associated with the multi-step method which is defined by**

$$y_{i+1}=a_{m-1}y_i+a_{m-2}y_{i-1}+\cdots+a_0y_{i+1-m}+h\cdot F(t_i,h,y_{i+1},y_i,\cdots,y_{i+1-m})$$

with $y_0=\alpha_0$, $y_1=\alpha_1$, $y_2=\alpha_2$, …, $y_{m-1}=\alpha_{m-1}$.

If $|\lambda_i|\leq 1$ for $\forall i=1, 2, \ldots, m$, and all roots with absolute value 1 are simple roots, then the multi-step method is stable

1. **Strongly stable: If $|\lambda_i|=1\Rightarrow\lambda_i=1$**
2. **Weakly stable: If $|\lambda_i|=1\Rightarrow\lambda_i$ may be other than unity.**

Eg. Is the following explicit multi-step method stable?

$$y_{i+1} = y_{i-3} + \frac{4h}{3}[2f(t_i, y_i) - f(t_{i-1}, y_{i-1}) + 2f(t_{i-2}, y_{i-2})]$$

(Sol.) $y_{i+1} = 0 \cdot y_i + 0 \cdot y_{i-1} + 0 \cdot y_{i-2} + 1 \cdot y_{i-3} + \frac{4h}{3}[\cdots]$

$\Rightarrow$ Characteristic polynomial is $p(\lambda)=\lambda^4-1=0$

$\lambda=1, i, -1, -i \Rightarrow |\lambda| \le 1 \Rightarrow$ stable, but $|\lambda|=1$ holds for $-1, i, -i \neq 1$. ∴ It is weakly stable!

Eg. Show that Adams-Bashforth method is strongly stable.

(Proof) $y_{i+1} = y_i + \frac{h}{24}[55f(t_i, y_i) - 59f(t_{i-1}, y_{i-1}) + 37f(t_{i-2}, y_{i-2}) - 9f(t_{i-3}, y_{i-3})]$

$\Rightarrow$ Characteristic polynomial is $p(\lambda)=\lambda^4-\lambda^3=0$

$\Rightarrow \lambda = 1,0,0,0 \Rightarrow |\lambda| \le 1 \Rightarrow$ stable

$|\lambda|=1$ holds only $\lambda=1$. ∴ It is strongly stable!