

Ecuaciones diferenciales ordinarias

Curso de Física Computacional

M. en C. Gustavo Contreras Mayén

Método Runge-Kutta de segundo orden.

1. Descripción.

Este programa calcula la solución de la EDO de segundo orden, mediante el método de RK2. Las constantes se definen en las instrucciones **DATA**, las condiciones iniciales están dadas en **YB** y **ZB**. Los resultados obtenidos se imprimen después de cada 50 pasos.

2. Variables.

Y, Z: y y z para un nuevo paso.

YB, ZB: y y z del paso anterior.

BM, KM: a y b de la ecuación a resolver.

3. Código.

```
1  PROGRAM masaresorte
2  REAL*8 M, K, K1, K2, L1, L2, KM
3  DATA T, K, M, B, Z, Y, H /0.0, 100.0, 0.5, 0.0, 0.0, 1.0, 0.001/
4  PRINT *, ' T Y Z '
5  PRINT 1, T, Y, Z
6 1  FORMAT (F10.5, 1P2E13.6)
7
8  KM=K/M
9  BM=B/M
10
11  OPEN(1,FILE='rk2masares.dat',STATUS='UNKNOWN')
12  DO N = 1, 20
13      DO KOUNT = 1, 50
14          T = T+H
15          K1 = H*Z
16          L1 = -H*(BM*Z+KM*Y)
17          K2 = H*(Z+L1)
18          L2 = - H*(BM*(Z+L1)+KM*(Y+K1))
19          Y = Y+(K1+K2)/2
20          Z = Z+(L1+L2)/2
21      END DO
22      WRITE(1,*)T, Y, Z
23      PRINT 1, T, Y, Z
24  END DO
25  CLOSE(1)
26  END PROGRAM masaresorte
```

Método de Runge-Kutta de cuarto orden.

1. **Descripción.** Se presenta un programa de Runge-Kutta de cuarto orden para resolver una ecuación diferencial de primer orden. Antes de ejecutar el programa, el usuario debe de definir la EDO a resolver, en el subprograma **FUN**. Cuando se ejecuta el programa, se le preguntará al usuario el número de pasos I , en el intervalo de impresión t , denotado TD . Entonces, el intervalo de tiempo se hace igual a $h = TD/I$. También se le pregunta al usuario, el máximo t en el que debe evaluarse la solución.

2. Variables

H : intervalo de tiempo, h

F: $f(y, t)$

K1, K2, K3, K4: k_1, k_2, k_3, k_4 , respectivamente

Y : y

YA : y en el subprograma que define la ecuación diferencial

X : t

XA : t de la ecuación diferencial en el subprograma

XL : valor máximo de t

TD : intervalo de impresión de t (la solución se imprime después de cada incremento de t por TD).

3. Código. A continuación se indica el código del programa.

```
1  PROGRAM rk4ecuacion
2      REAL K1, K2, K3, K4
3
4      PRINT *
5 1  PRINT *, 'Esquema de Runge-Kutta de cuarto orden'
6      PRINT *
7      PRINT *, 'Intervalo de impresion de T?'
8      READ *, XPR
9      PRINT *, 'Numero de pasos en un intervalo de impresion?'
10     READ *, I
11     PRINT *, 'Maximo?'
12     READ *, XL
13
14 !  AQUÍ SE FIJA EL VALOR INICIAL DE LA SOLUCION
15     Y=473
16
17 !  H ES EL INTERVALO DE TIEMPO
18     H=XPR/I\end{enumerate}
19
20     PRINT *, 'H= ', H
21
22     OPEN (1, FILE='datostemp.dat', STATUS='REPLACE')
23 !  SE INICIALIZA EL TIEMPO
24     XP=0
25     HH=H/2
26     PRINT *
27     PRINT *, '_____',
28     PRINT *, ' T      Y'
29     PRINT *, '_____',
30     PRINT *, XP, Y
31     82 FORMAT (1X, F10.6, 7X, 1PE15.6)
32
33 !  AVANZA I PASOS EN CADA INTERVALO DE IMPRESION
34
35 30  DO J= 1, I
36      XB=XP
37      XP=XP+H
38      YN=Y
39      XM=XB+HH
40      K1=H*FUN(YB, XB)
41      K2=H*FUN(YN+K1/2, XM)
42      K3=H*FUN(YN+K2/2, XM)
43      K4=H*FUN(YN+K3, XP)
44      Y=YN+(K1+K2*2+K3*2+K4)/6
45      END DO
46
47      PRINT 82, XP, Y
48      WRITE(1, *) XP, Y
49      IF (XP .LE. XL) GOTO 30
```

```

50      PRINT *
51      PRINT *, 'Se ha excedido del limite de X'
52      PRINT *
53      PRINT *
54
55      CLOSE(1)
56      PRINT *, 'Oprime 1 para continuar, 0 para terminar'
57      READ *, K
58      IF (K .EQ. 1) GOTO 1
59      PRINT *
60      END PROGRAM rk4ecuacion
61      ! ++++++
62      FUNCTION FUN(Y,X)
63          rho = 300
64          volumen=0.001
65          area = 0.25
66          caloresp = 900
67          ctransferencia = 30
68          emisividad = 0.8
69          ctestefanbol = 5.67E-8
70
71          constante1 = area/(rho*caloresp*volumen)
72          constante2 = emisividad*ctestefanbol
73
74          FUN=X*(constante1*(constante2*(bignumerote-Y**4)+ctransferencia*(297-Y)))
75          RETURN
76      END|
77      !+++++
78      FUNCTION bignumerote()
79          integer, parameter:: DD = selected_int_kind(15)
80          integer(kind = DD):: bignum
81          bignum = 297_DD**4
82          RETURN
83      END

```

Parámetro KIND.

Especifica el tipo de dato. Si no se especifica, se asume el tipo de dato por defecto. Por ejemplo, Fortran 90 provee tres tipos de datos de parámetro KIND para datos de tipo real:

REAL(KIND=4) (o REAL*4)

REAL(KIND=8) (o REAL*8)

REAL(KIND=16) (o REAL*16)

Función: Selected Int Kind(R)

Devuelve el tipo de dato entero adecuado para representar todos los valores n en el rango

$$-10^R < n < 10^R$$

Si no es posible ese tipo de dato, devuelve el valor de -1 .

La aplicación del método RK4 a un conjunto de ecuaciones diferenciales es análoga del método de segundo orden. Sean un conjunto de dos ecuaciones:

$$\begin{aligned} y' &= f(y, z, t) \\ z' &= g(y, z, t) \end{aligned} \tag{1}$$

El método RK4 para este conjunto es:

$$\begin{aligned}
 k_1 &= hf(y_n, z_n, t_n) \\
 l_1 &= hg(y_n, z_n, t_n) \\
 k_2 &= hf\left(y_n + \frac{k_1}{2}, z_n + \frac{l_1}{2}, t_n + \frac{h}{2}\right) \\
 l_2 &= hg\left(y_n + \frac{k_1}{2}, z_n + \frac{l_1}{2}, t_n + \frac{h}{2}\right) \\
 k_3 &= hf\left(y_n + \frac{k_2}{2}, z_n + \frac{l_2}{2}, t_n + \frac{h}{2}\right) \\
 l_3 &= hg\left(y_n + \frac{k_2}{2}, z_n + \frac{l_2}{2}, t_n + \frac{h}{2}\right) \\
 k_4 &= hf(y_n + k_3, z_n + l_3, t_n + h) \\
 l_4 &= hg(y_n + k_3, z_n + l_3, t_n + h) \\
 y_{n+1} &= y_n + \frac{1}{6} [k_1 + 2k_2 + 2k_3 + k_4] \\
 z_{n+1} &= z_n + \frac{1}{6} [l_1 + 2l_2 + 2l_3 + l_4]
 \end{aligned}
 \tag{2}$$

Incluso cuando el número de ecuaciones en un conjunto es mayor que dos, el método RK4 es esencialmente el mismo.

Ejemplo: Resuelve el siguiente conjunto de EDO mediante el esquema RK4, utilizando $h = 0,2\pi$ y $h = 0,05\pi$

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

1. **Descripción:** Este programa resuelve un conjunto de cualquier número de EDO de primer orden. En la subrutina **FUNCT** se define el conjunto de ecuaciones diferenciales a resolver. En el programa principal se define el número de ecuaciones **IM**, así como los valores de las condiciones iniciales. Para correr el programa con un nuevo problema, el usuario debe de cambiar las ecuaciones en **FUNCT**, el valor de **IM** y las condiciones iniciales.

2. Variables:

Y(1) : y

Y(2) : z

Y(I) : i-ésima incógnita

YN(I) : y_n para $I = 1$ y z_n para $I = 2$, etc.

YA(I) : $y_n + \frac{k_1}{2}$ o $y_n + \frac{k_2}{2}$ o $y_n + k_3$ para $I = 1$; $z_n + \frac{l_1}{2}$ o $z_n + \frac{l_2}{2}$ o $z_n + l_3$ para $I = 2$

EK(J,1), J=1,2,3,4 : k_1, k_2, k_3, k_4

EK(J,2), J=1,2,3,4 : l_1, l_2, l_3, l_4

EK(J,M), J=1,2,3,4 : similar a la anterior para la m-ésima EDO

IM : número de ecuaciones en el conjunto

NS : número de intervalos de tiempo en un intervalo de impresión

XP : límite máximo de t

TD : intervalo de impresión para t

```

1  PROGRAM RK4.SISTEMA
2  DIMENSION YA(0:10) , YN(0:10) , EK(0:4,0:10) , Y(0:10)
3  PRINT *
4  PRINT *, "          ESQUEMA DE RK4 PARA UN CONJUNTO DE ECUACIONES"
5  PRINT *
6  !numero de ecuaciones
7  IM = 2

```

```

8
9  Y(1) = 1  !condicion inicial para y1 en t=0
10 Y(2) = 0  !condicion inicial para y2 en t=0
11
12 1 PRINT *
13 PRINT *, 'INTERVALO DE IMPRESION DE T?'
14 READ *, TD
15
16 PRINT *, 'NUMERO DE PASOS EN UN INTERVALO DE IMPRESION DE T?'
17 READ *, NS
18
19 PRINT *, 'T MAXIMO PARA DETENER LOS CALCULOS?'
20 READ *, XL
21
22 H = TD/NS
23
24 PRINT *, 'H = ', H
25
26 XP = 0
27 HH = H/2
28
29 PRINT *
30
31 LI = 0  !inicializacion del numero de linea
32 PRINT *, 'LINEA      T      Y(1),      Y(2), ..... '
33 WRITE(*,98) LI, XP, (Y(I),I=1,IM)
34
35 28 LI = LI + 1
36 DO N = 1, NS
37   XB = XP      !tiempo anterior
38   XP = XP + H  !tiempo nuevo
39   XM = XB + HH !tiempo en el punto medio
40
41   J = 1      !se calcula k1
42   DO I = 1, IM
43     YA(I) = Y(I)
44   END DO
45   XA = XB
46   CALL FUNCT(EK,J,YA,H)
47
48   J = 2      !se calcula k2
49   DO I = 1, IM
50     YA(I) = Y(I) + EK(1,I)/2
51   END DO
52   XA = XM
53   CALL FUNCT(EK,J,YA,H)
54
55   J = 3      !se calcula k3
56   DO I = 1, IM
57     YA(I) = Y(I) + EK(2,I)/2
58   END DO
59   XA = XM
60   CALL FUNCT(EK,J,YA,H)
61
62   J = 4      !se calcula k4
63   DO I = 1, IM
64     YA(I) = Y(I) + EK(3,I)/2
65   END DO
66   XA = XP
67   CALL FUNCT(EK,J,YA,H)
68
69   DO I = 1, IM !esquema RK4
70     Y(I) = Y(I) + (EK(1,I) + EK(2,I)*2 + EK(3,I)*2 + EK(4,I))/6
71   END DO
72 END DO

```

```

73
74   WRITE(*,98) LI,XP,(Y(I),I=1,IM)
75 98  FORMAT(1X,I2,F10.6,2X,1P4E16.6/(15X,1P4E16.6))
76   IF (XP .LT. XL) GOTO 28
77 200 PRINT *
78   PRINT *, 'TECLEA 1 PARA CONTINUAR O 0 PARA TERMINAR'
79   READ *, K
80   IF (K .EQ. 1) GOTO 1
81   PRINT *
82   END PROGRAM
83 !+++++
84   SUBROUTINE FUNCT(EK,J,YA,H)
85   DIMENSION EK(0:4,0:10), YA(0:10)
86
87   EK(J,1) = YA(2)*H
88   EK(J,2) = -YA(1)*H
89   RETURN
90   END

```