METODOS NUMERICOS

Matemática Intermedia III

Ingenieria Usac
Ing. Luis Carlos Bolaños Mendez

METODO DE EULER

Este es uno de los métodos mas simples para aproximar soluciones de problemas de valores inciales de ecuaciones de la forma y'=f(x,y) sujeta a las condiciones y(x0)=y0 la formula del método de Euler es la siguiente
Yn+1=Yn+h f(x,y)

In[1]:=
```
euler1[] := Module[{n, i, x},
  f[x_, y_] = Input["ingrese la funcion"];
  h = Input["ingrese el paso"];
  nn = Input["Ingrese el numero de iteraciones"];
  X = Input["ingrese en una lista los valores iniciales"];
  Yn = Range[nn + 1];
  Yn[[1]] = X[[2]];
  Xn = X[[1]];
  For[i = 2, i ≤ nn + 1, i++, Yn[[i]] = Yn[[i - 1]] + h * f[Xn, Yn[[i - 1]]];
   Xn = Xn + h];
  Xn = Table[X[[1]] + n * h, {n, 0, nn, 1}];
  Xn = {"xn"} ~ Join ~ Xn;
  Yn = {"yn"} ~ Join ~ Yn;
  Transpose[{Xn, Yn}] // TableForm
 ]
```

EJEMPLO
Muestre una aproximacion de y(1.5) utilizando el método de Euler para y=2 x y con y(1)=1, utilice h=0.05

In[3]:=
```
euler1[]
```

Out[3]//TableForm=

xn	yn
1.	1
1.05	1.1
1.1	1.2155
1.15	1.34921
1.2	1.50436
1.25	1.68489
1.3	1.8955
1.35	2.14191
1.4	2.43107
1.45	2.77142
1.5	3.17328
1.55	3.64927

METODO DE EULER MEJORADO

Este método hace una mejora y reduce el error. las formulas son las siugientes

$Yn+1 = Yn + h\,(f\,(Xn, Yn) + f\,(Xn+1, Y^*n+1))/2$

$Y^*n+1=Yn+hf(xn,yn)$

In[4]:=
```
euler2[] := Module[{n, i, x},
  f[x_, y_] = Input["ingrese la funcion"];
  h = Input["ingrese el paso"];
  nn = Input["Ingrese el numero de iteraciones"];
  X = Input["ingrese en una lista los valores iniciales"];
  Yn = Range[nn + 1];
  Yn1 = Range[nn + 1];
  Yn[[1]] = X[[2]];
  Xn = X[[1]];
  For[i = 2, i ≤ nn + 1, i++, Yn1[[i]] = Yn[[i - 1]] + h * f[Xn, Yn[[i - 1]]];
   Yn[[i]] = Yn[[i - 1]] + h / 2 * (f[Xn, Yn[[i - 1]]] + f[Xn + h, Yn1[[i]]]);
   Xn = Xn + h];
  Xn = Table[X[[1]] + n * h, {n, 0, nn, 1}];
  Xn = {"Xn"} ~ Join ~ Xn;
  Yn1 = {"Yn*"} ~ Join ~ Yn1;
  Yn = {"Yn"} ~ Join ~ Yn;
  Transpose[{Xn, Yn1, Yn}] // TableForm
 ]
```

EJEMPLO

Muestre una aproximacion de y(1.5) utilizando el método de Euler Mejorado para y=2 x y con y(1)=1, utilice h=0.05

In[5]:=
```
euler2[]
```

Out[5]//TableForm=

Xn	Yn*	Yn
1.	1	1
1.05	1.1	1.10775
1.1	1.22406	1.23323
1.15	1.36889	1.37977
1.2	1.53844	1.55141
1.25	1.73758	1.7531
1.3	1.97223	1.99086
1.35	2.24967	2.27212
1.4	2.57885	2.60601
1.45	2.97085	3.00381
1.5	3.43937	3.47954
1.55	4.00147	4.05062

METODO DE RUNGE KUTTA DE CUARTO ORDEN

En este metodo se calcualn 4 valores y luego se hace un promedio ponderado de estos 4 valores las formulas son las siguientes.

Yn+1=yn+h/6(k1+2K2+2K3+K4)

K1=f(Xn,Yn)

K2=f(Xn+1/2h,Yn+1/2*h*K1)

K3=f(Xn+1/2h,Yn+1/2*h*K2)

K4=f(Xn+h,Yn+hK3)

In[7]:=
```
rk4[] := Module[{i, n},
    f[x_, y_] = Input["ingrese la funcion"];
    h = Input["ingrese el paso"];
    nn = Input["Ingrese el numero de iteraciones"];
    X = Input["ingrese en una lista los valores iniciales"];
    Yn = Range[nn + 1];
    Yn[[1]] = X[[2]];
    Xn = X[[1]];
    K1 = Range[nn + 1];
    K2 = Range[nn + 1];
    K3 = Range[nn + 1];
    K4 = Range[nn + 1];
    For[i = 2, i ≤ nn + 1, i++, K1[[i]] = f[Xn, Yn[[i - 1]]];
     K2[[i]] = f[Xn + h / 2, Yn[[i - 1]] + 1 / 2 * h * K1[[i]]];
     K3[[i]] = f[Xn + h / 2, Yn[[i - 1]] + 1 / 2 * h * K2[[i]]];
     K4[[i]] = f[Xn + h, Yn[[i - 1]] + h * K3[[i]]];
     Yn[[i]] = Yn[[i - 1]] + h / 6 * (K1[[i]] + 2 K2[[i]] + 2 K3[[i]] + K4[[i]]);
     Xn = Xn + h];
    Xn = Table[X[[1]] + n * h, {n, 0, nn, 1}];
    Xn = {"Xn"} ~Join~ Xn;
    K1 = {"K1"} ~Join~ K1;
    K2 = {"K2"} ~Join~ K2;
    K3 = {"K3"} ~Join~ K3;
    K4 = {"K4"} ~Join~ K4;
    Yn = {"Yn"} ~Join~ Yn;
    Transpose[{Xn, K1, K2, K3, K4, Yn}] // TableForm
   ];
```

EJEMPLO

Muestre una aproximacion de y(1.5) utilizando el método de Runge Kutta de cuarto orden para y=2 x y con y(1)=1, utilice h=0.05

In[8]:=
```
rk4[]
```

Out[8]//TableForm=

Xn	K1	K2	K3	K4	Yn
1.	1	1	1	1	1
1.05	2	2.1525	2.16032	2.32683	1.10794
1.1	2.32667	2.50712	2.51682	2.71431	1.23368
1.15	2.71409	2.92844	2.9405	3.17562	1.38057
1.2	3.17532	3.4309	3.44592	3.72689	1.55271
1.25	3.7265	4.03238	4.05111	4.38816	1.75505
1.3	4.38763	4.7551	4.77852	5.18435	1.99371
1.35	5.18366	5.62676	5.65611	6.1466	2.27618
1.4	6.14569	6.68201	6.71888	7.31395	2.61169
1.45	7.31274	7.96436	8.01078	8.73547	3.01168
1.5	8.73387	9.52858	9.58719	10.4731	3.49033