

1 Introduccion

El método de Euler es un método de primer orden, lo que significa que el error local es proporcional al cuadrado del tamaño del paso, y el error global es proporcional al tamaño del paso. El método de Euler regularmente sirve como base para construir métodos más complejos.

2 Euler

Considere el problema de calcular la pendiente de una curva desconocida que comienza en un punto dado y satisface una cierta ecuación diferencial dada. Se puede pensar en la ecuación diferencial como una fórmula que nos permite calcular la pendiente de la recta tangente a la curva en cualquier punto de la curva, una vez que el punto ha sido calculado.

La idea es que a pesar de que la curva es desconocida en un principio, su punto de comienzo, al cual denotamos por A_0 , es conocido. Entonces, de la ecuación diferencial se puede calcular la pendiente de la curva en el punto A_0 y por lo tanto la recta tangente a la curva.

Ahora, dando un pequeño paso sobre dicha recta, podemos tomarnos un nuevo punto A_1 y suponer que dicho punto pertenece a la curva, entonces seguimos el mismo razonamiento aplicado anteriormente y volvemos a calcular la pendiente de la recta tangente a la curva en el punto A_1 . Luego de varios pasos tendremos formada una curva poligonal $A_0A_1A_2A_3...$. En general esta curva que obtenemos al aplicar el método no diverge lejos de la curva original, además el error entre ambas curvas se puede minimizar si se dan pasos muy pequeños al avanzar sobre la recta tangente a la curva y además el intervalo sobre el que trabajamos es finito (aunque las cosas son más complicadas para ecuaciones inestables, como se discute más abajo)...

```
program main

real :: h, t, f, l
real,dimension(2) :: w,x
integer :: i

    print*,"Longitud pendulo"
    read*,l
    print*,"Angulo inicial << 1"
    read*,x(1)

    open(1,file='tabla.dat',status='unknown')
    t=0
    w(1)=0
    h=(2.0*3.1416*sqrt(l/9.81))/100
```

```

do i = 1,100
  x(2)=x(1) + h*w(1)
  w(2)=w(1) + h*f(x(1),l)
  write(1,*) t,x(2)
  t = t + h
  x(1)=x(2)
  w(1)=w(2)

end do

end program main

function f(x,l)
real, intent(in)::x, l
f = -(9.81)*x/l
end function f

```