# Punto 1

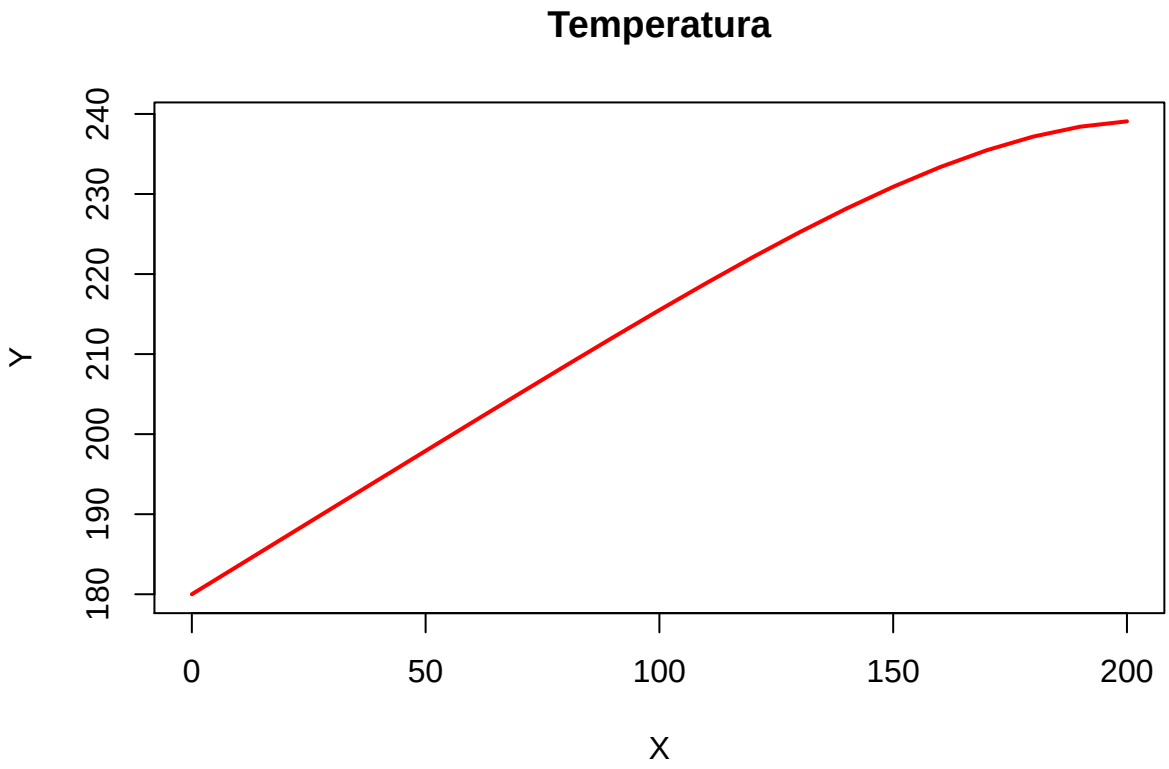
David Leonardo Gomez  Ericka Julieth Torres  Leyder Fabian Vera

28 Octubre 2018

```
#install.packages("pracma")
library(pracma)

metodoEuler <- function(f, h, xi, yi, xf)
{
  N = (xf - xi) / h
  x = y = numeric(N+1)
  x[1] = xi;
  y[1] = yi;
  i = 1
  while (i <= N)
  {
    x[i+1] = x[i]+h
    y[i+1] = y[i]+(h*f(x[i],y[i]))
    i = i+1
  }
  return (data.frame(X = x, Y = y))
}

f <- function(x, y) {(-5.6e-8*0.8*0.5*(x^4-200^4)) / 100}

e1 = metodoEuler(f, 10, 0, 180, 200)

e1[nrow(e1),]
```

```
##      X        Y
## 21 200 239.0763
```

```
plot(e1$X,e1$Y, type='l', lwd=2, col='red',
     main="Temperatura",xlab="X", ylab="Y")
```


Temperatura
Y
180
190
200
210
220
230
240
0
50
100
150
200
X

# Punto 2

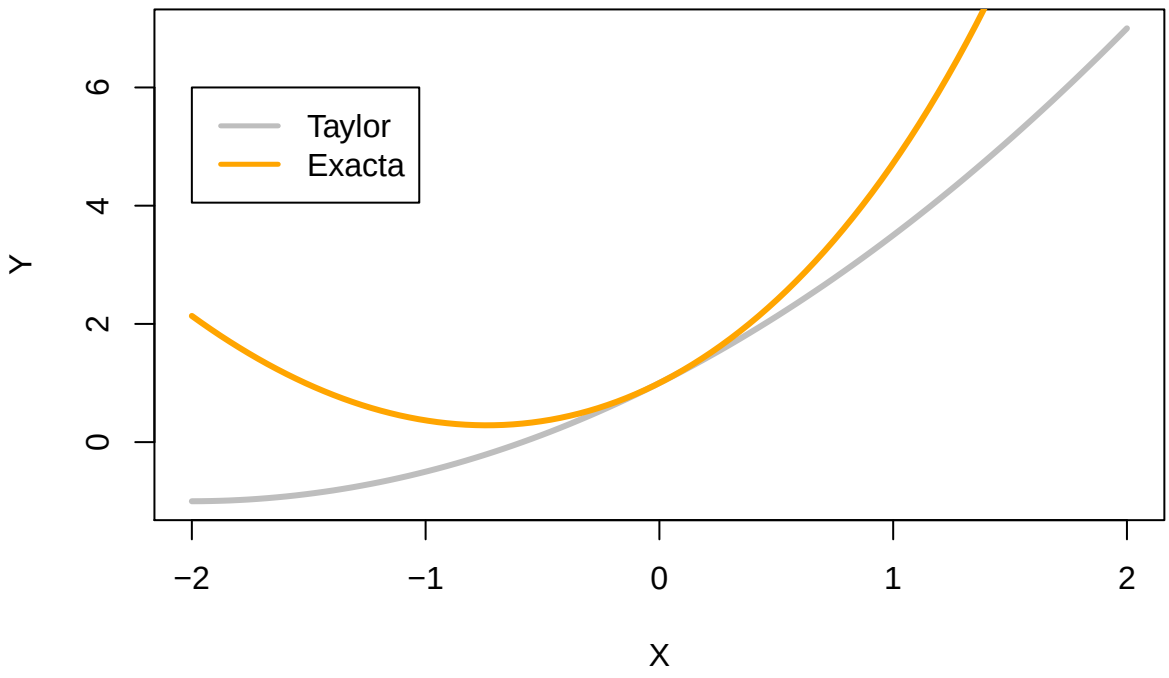
```
#install.packages("pracma")
options(digits=8)
library(pracma)
exact <- function(x)
{
  return(exp(x)*(((x^2)*exp(-x))+(x*exp(-x))+1))
}
f <- function(x,y)
{
  return((2*x)-((x^3)/3)+((x^2)/2)+1)
}
pT = taylor(f, 0, n = 2)
print("Valores con aproximacion de Taylor: ")
```

```
## [1] "Valores con aproximacion de Taylor: "
```

```
x <- seq(0, 1, length.out=5)
print(data.frame(X = x, Y = polyval(pT,x)))
```

```
##      X       Y
## 1 0.00 1.00000
## 2 0.25 1.53125
## 3 0.50 2.12500
## 4 0.75 2.78125
## 5 1.00 3.50000
```

```
x <- seq(-2, 2, length.out=100)
pTa <- polyval(pT, x)
plot(x, pTa, type = "l", col = "gray",main="Comparación",xlab="X",ylab="Y", lwd = 3)
lines(x, exact(x), col = "orange" ,lwd = 3)
legend(-2,6,c('Taylor','Exacta'),lty=c(1,1),lwd=c(2.5,2.5),col=c('gray','orange'))
```




```
cat("\n")
```

```
print("con h= 0.1:")
```

```
## [1] "con h= 0.1:"
```

```
cat("valor real       valor con taylor     error\n")
```

```
## valor real       valor con taylor     error
```

```
cat(exact(0.1),"      ",polyval(pT,0.1),"             ",(exact(0.1)-polyval(pT,0.1))/exact(0.1))
```

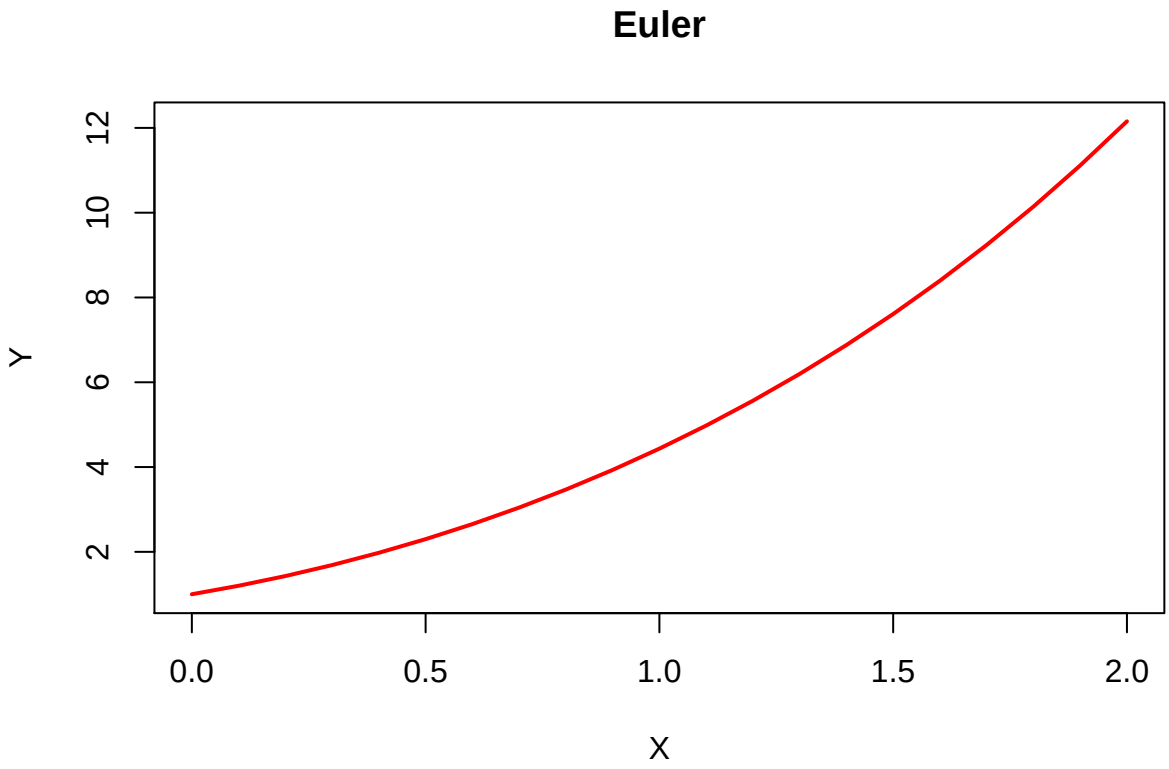
```
## 1.2151709        1.205               0.0083699486
```

# Punto 3

```r
#install.packages("pracma")
library(pracma)

f <- function(x, y) {x+y+1-x^2}

metodoEuler <- function(f, h, xi, yi, xf)
{
  N = (xf - xi) / h
  x = y = numeric(N+1)
  x[1] = xi;
  y[1] = yi;
  i = 1
  while (i <= N)
  {
    x[i+1] = x[i]+h
    y[i+1] = y[i]+(h*f(x[i],y[i]))
    i = i+1
  }
  return (data.frame(X = x, Y = y))
}

e1 = metodoEuler(f, 0.1, 0, 1, 2)

e1[nrow(e1),]
```

```
##    X        Y
## 21 2 12.15475
```

```r
plot(e1$X,e1$Y, type='l', lwd=2, col='red',
     main="Euler",xlab="X", ylab="Y")
```


Euler
Y
2
4
6
8
10
12
0.0
0.5
1.0
1.5
2.0
X

# Punto 4

```
f<-function(fcn,x,y){
  return(eval(fcn))
}

calcular <- function(x,y){
  h <- 0.1
  m <- 20
  i <- x

  for(i  in 1:m){
    k1 <- f(expression(x+y+1-(x*x)),x[i],y[i])
    k2 <- f(expression(x+y+1-(x*x)),(x[i] + h),(y[i] + k1))
    y[i+1] <- y[i] + 0.5 *(k1 + k2)
    x[i+1] <- x[i] + h
    cat("X:", x[i], "Y: ", y[i], "K1 : " , k1, "K2:" , k2, "\n")
  }
}

calcular(0,1)
```

```
## X: 0 Y:  1 K1 :  2 K2: 4.09
## X: 0.1 Y:  4.045 K1 :  5.135 K2: 10.34
## X: 0.2 Y:  11.7825 K1 :  12.9425 K2: 25.935
## X: 0.3 Y:  31.22125 K1 :  32.43125 K2: 64.8925
## X: 0.4 Y:  79.88312 K1 :  81.12313 K2: 162.2562
## X: 0.5 Y:  201.5728 K1 :  202.8228 K2: 405.6356
## X: 0.6 Y:  505.802 K1 :  507.042 K2: 1014.054
## X: 0.7 Y:  1266.35 K1 :  1267.56 K2: 2535.07
## X: 0.8 Y:  3167.665 K1 :  3168.825 K2: 6337.58
## X: 0.9 Y:  7920.868 K1 :  7921.958 K2: 15843.83
## X: 1 Y:  19803.76 K1 :  19804.76 K2: 39609.41
## X: 1.1 Y:  49510.84 K1 :  49511.73 K2: 99023.34
## X: 1.2 Y:  123778.4 K1 :  123779.1 K2: 247558.1
## X: 1.3 Y:  309447 K1 :  309447.6 K2: 618895.1
## X: 1.4 Y:  773618.4 K1 :  773618.8 K2: 1547237
## X: 1.5 Y:  1934047 K1 :  1934047 K2: 3868093
## X: 1.6 Y:  4835117 K1 :  4835117 K2: 9670233
## X: 1.7 Y:  12087791 K1 :  12087791 K2: 24175582
## X: 1.8 Y:  30219478 K1 :  30219478 K2: 60438955
## X: 1.9 Y:  75548694 K1 :  75548694 K2: 151097387
```

# Punto 5

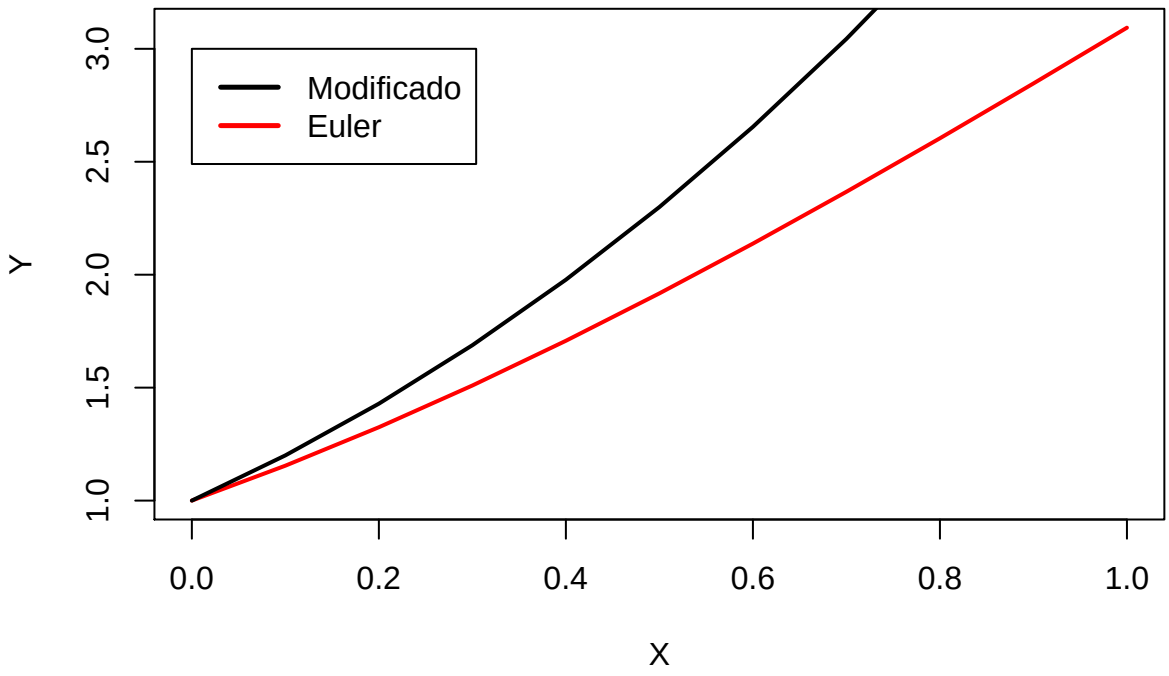
```r
#install.packages("pracma")
library(pracma)

# funcion, h, y(xi) = yi, x final
metodoEulerMod <- function(f, h, xi, yi, xf)
{
  N = (xf - xi) / h
  x = y = numeric(N+1)
  x[1] = xi;
  y[1] = yi;
  i = 1
  while (i <= N)
  {
    x[i+1] = x[i]+h
    y[i+1] = y[i]+(h/2*(f(x[i],y[i])+f(x[i+1],y[i+1])))
    i = i+1
  }
  return (data.frame(X = x, Y = y))
}

f <- function(x, y) {x+y+1-x^2}

e2 = metodoEulerMod(f, 0.1 , 0, 1, 1)

e2[nrow(e2),]
```

```
##    X        Y
## 11 1 3.093433
```

```r
plot(e2$X,e2$Y, type='l', lwd=2, col='red',
     xlab="X", ylab="Y")

metodoEuler <- function(f, h, xi, yi, xf)
{
  N = (xf - xi) / h
  x = y = numeric(N+1)
  x[1] = xi;
  y[1] = yi;
  i = 1
  while (i <= N)
  {
    x[i+1] = x[i]+h
    y[i+1] = y[i]+(h*f(x[i],y[i]))
    i = i+1
  }
  return (data.frame(X = x, Y = y))
}

e1 = metodoEuler(f, 0.1, 0, 1, 1)

e1[nrow(e1),]
```

```
##    X        Y
## 11 1 4.434368
```

```r
lines(e1$X,e1$Y, type='l', lwd=2, col='black',
      main="Euler",xlab="X", ylab="Y")

legend(0,3, # places a legend at the appropriate place
       c('Modificado','Euler'), # puts text in the legend
       lty=c(1,1), # gives the legend appropriate symbols (lines)
       lwd=c(2.5,2.5),col=c('black','red'))
```



# Punto 7

```
list.of.packages <- c("phaseR")
new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]
if(length(new.packages)) install.packages(new.packages)

library(phaseR)
library(pracma)

f<-function(fcn,x,y){
  return(eval(fcn))
}

metodoEuler <- function(f, h, xi, yi, xf)
{
  N = (xf - xi) / h
  x = y = numeric(N+1)
  x[1] = xi;
  y[1] = yi;
  i = 1
  while (i <= N)
  {
    x[i+1] = x[i]+h
    y[i+1] = y[i]+(h*f(x[i],y[i]))
    i = i+1
  }
  return (data.frame(X = x, Y = y))
}

e1 = metodoEuler(f, 0.1, 0, 1, 2)

e1[nrow(e1),]
```

```
##    X   Y
## 21 2 2.9
```

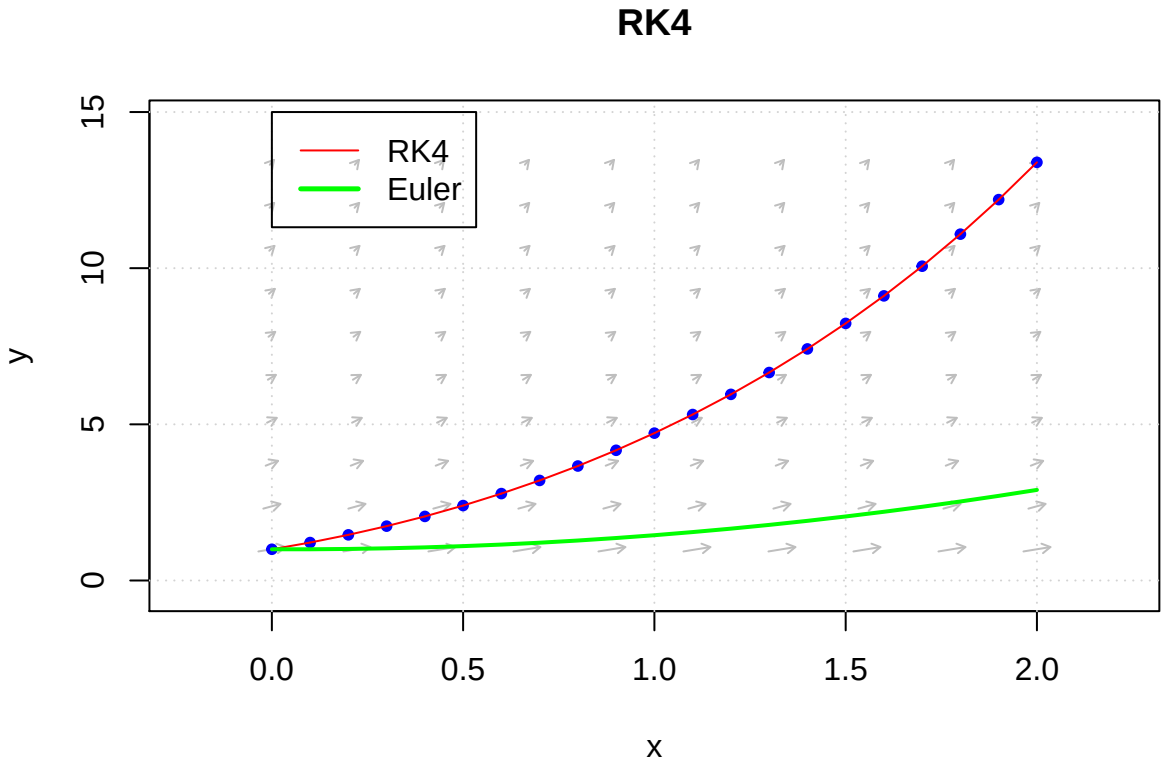
```
# Solo para prueba con dy=x+y, y(0)=1
obtenerErrorAbsoluto<-function(x,y){
  solucion=exp(x)*((-x*exp(-x))-exp(-x)+2)
  return(abs(y-solucion))
}

graficarCampoPendiente<-function(x0, xn, y0, yn, fcn, numpendientes, metodo){
  apma1 <- function(t, y, parameters){
    a <- parameters[1]
    dy <- a*(f(fcn, t, y))
    list(dy)
  }
  apma1.flowField <- flowField(apma1, x = c(x0, xn),
                               y   = c(y0, yn), parameters = c(1),
                               points = numpendientes, system = "one.dim",
                               add = FALSE, xlab = "x", ylab = "y",
                               main = metodo)
```

```
  grid()
}

graficarSolucionNumerica<-function (x, y){
  points (x, y, pch=20, col="blue")
  for (i in 2:length(x)){
    segments(x[i-1], y[i-1], x[i], y[i], col="red")
  }
}

rk4<-function(dy, ti, tf, y0, h, graficar=TRUE, numpendientes=10){
  t<-seq(ti, tf, h)
  y<-c(y0)
  cat("x    |y         |k1        |k2        |k3        |k4       |error absoluto\n")
  for(i in 2:length(t)){
    k1=h*f(dy, t[i-1], y[i-1])
    k2=h*f(dy, t[i-1]+h/2, y[i-1]+k1*(0.5))
    k3=h*f(dy, t[i-1]+h/2, y[i-1]+k2*(0.5))
    k4=h*f(dy, t[i-1]+h, y[i-1]+k3)
    y<-c(y, y[i-1]+1/6*(k1+2*k2+2*k3+k4))
    cat(t[i-1]," | ", y[i-1]," | ",k1," | ",k2," | ",k3," | ",k4," | ",obtenerErrorAbsoluto(t[i-1],y[i-
  }
  if (graficar){
    graficarCampoPendiente(min(t), max(t), min(y), max(y), dy, numpendientes, "RK4")
    graficarSolucionNumerica(t, y)
  }
  rta<-list(w=y, t=t)
}

rk3<-function(dy, ti, tf, y0, h, graficar=TRUE, numpendientes=10){
  t<-seq(ti, tf, h)
  y<-c(y0)
  cat("x    |y         |k1         |k2        |k3       |error absoluto\n")
  for(i in 2:length(t)){
    k1=h*f(dy, t[i-1], y[i-1])
    k2=h*f(dy, t[i-1]+h/2, y[i-1]+k1*(0.5))
    k3=h*f(dy, t[i-1]+h, y[i-1]-k1+2*k2)
    y<-c(y, y[i-1]+1/6*(k1+4*k2+k3))
    cat(t[i-1]," | ", y[i-1]," | ",k1," | ",k2," | ",k3," | ",obtenerErrorAbsoluto(t[i-1],y[i-1]),"\n")
  }
  if (graficar){
    graficarCampoPendiente(min(t), max(t), min(y), max(y), dy, numpendientes, "RK3")
    graficarSolucionNumerica(t, y)
  }
  rta<-list(w=y, t=t)
}

r<-rk4(expression(x+y+1 -(x*x)), 0, 2, 1, 0.1)
```

```
## x    |y         |k1        |k2        |k3        |k4       |error absoluto
## 0  |  1  |  0.2  |  0.21475  |  0.2154875  |  0.2305488  |  0
## 0.1  |  1.215171  |  0.2305171  |  0.2457929  |  0.2465567  |  0.2621727  |  0.1048288
## 0.2  |  1.461402  |  0.2621402  |  0.2779972  |  0.2787901  |  0.2950192  |  0.2185966
## 0.3  |  1.739858  |  0.2949858  |  0.3114851  |  0.31231  |  0.3292168  |  0.3401402
```

```
## 0.4  |  2.051823  |  0.3291823  |  0.3463914  |  0.3472519  |  0.3649075  |  0.4681739
## 0.5  |  2.398719  |  0.3648719  |  0.3828655  |  0.3837652  |  0.4022485  |  0.6012768
## 0.6  |  2.782116  |  0.4022116  |  0.4210722  |  0.4220152  |  0.4414132  |  0.7378787
## 0.7  |  3.20375  |  0.441375  |  0.4611937  |  0.4621846  |  0.4825934  |  0.8762442
## 0.8  |  3.665537  |  0.4825537  |  0.5034314  |  0.5044753  |  0.5260012  |  1.014455
## 0.9  |  4.169599  |  0.5259599  |  0.5480078  |  0.5491102  |  0.5718709  |  1.150392
## 1  |  4.718276  |  0.5718276  |  0.595169  |  0.5963361  |  0.6204612  |  1.281713
## 1.1  |  5.31416  |  0.620416  |  0.6451867  |  0.6464253  |  0.6720585  |  1.405827
## 1.2  |  5.960109  |  0.6720109  |  0.6983615  |  0.699679  |  0.7269788  |  1.519875
## 1.3  |  6.659288  |  0.7269288  |  0.7550252  |  0.75643  |  0.7855718  |  1.620694
## 1.4  |  7.41519  |  0.785519  |  0.8155449  |  0.8170462  |  0.8482236  |  1.70479
## 1.5  |  8.231677  |  0.8481677  |  0.8803261  |  0.881934  |  0.9153611  |  1.768299
## 1.6  |  9.113019  |  0.9153019  |  0.9498169  |  0.9515427  |  0.9874561  |  1.806954
## 1.7  |  10.06393  |  0.9873931  |  1.024513  |  1.026369  |  1.06503  |  1.816037
## 1.8  |  11.08963  |  1.064963  |  1.104961  |  1.106961  |  1.148659  |  1.790334
## 1.9  |  12.19587  |  1.148587  |  1.191767  |  1.193926  |  1.23898  |  1.724085
```

```r
lines(e1$X,e1$Y, type='l', lwd=2, col='green',
      main="Euler",xlab="X", ylab="Y")
legend(0,15, # Posicion de la leyenda
       c('RK4','Euler'), # Nombre de lineas
       lty=c(1,1), # Pone los simbolos de prueba (lineas)
       lwd=c(1,2.5),col=c('red','green'))
```

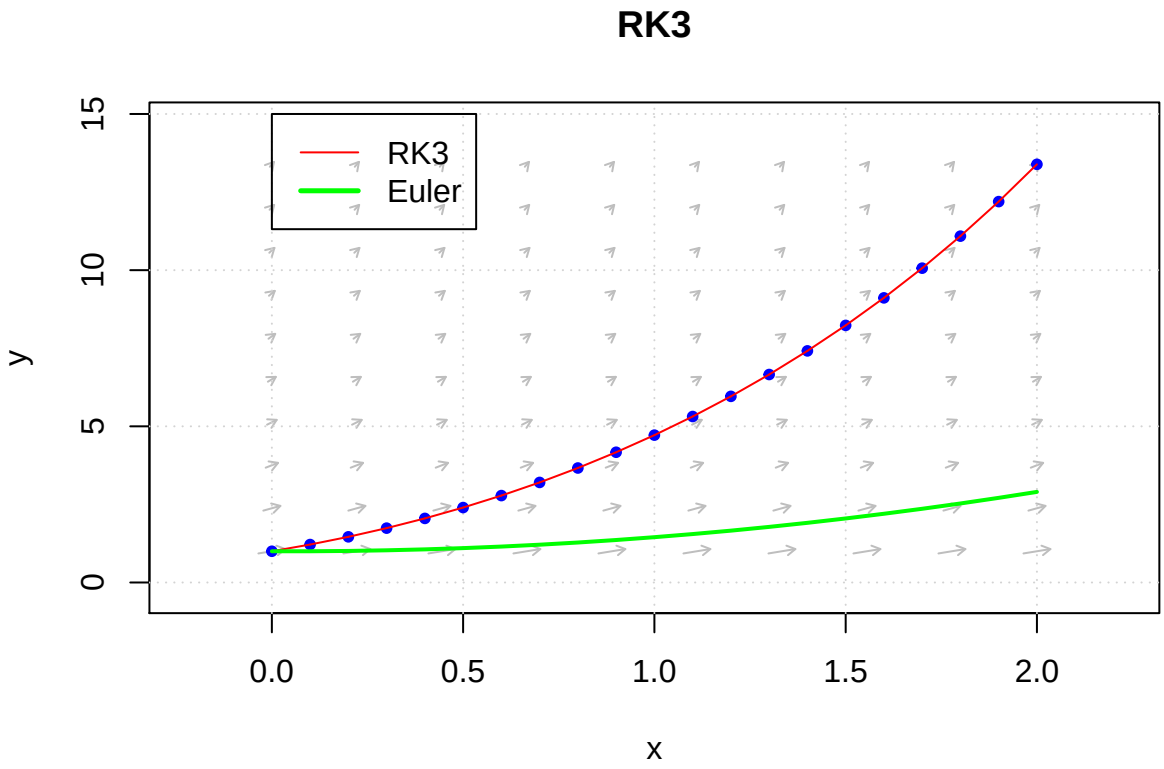



```r
r2<-rk3(expression(x+y+1 -(x*x)), 0, 2, 1, 0.1)
```

```
## x    |y         |k1         |k2         |k3         |error absoluto
```

```
## 0  |  1  |  0.2  |  0.21475  |  0.23195  |  0
## 0.1  |  1.215158  |  0.2305158  |  0.2457916  |  0.2636226  |  0.1048165
## 0.2  |  1.461376  |  0.2621376  |  0.2779945  |  0.2965227  |  0.2185703
## 0.3  |  1.739816  |  0.2949816  |  0.3114806  |  0.3307795  |  0.3400979
## 0.4  |  2.051763  |  0.3291763  |  0.3463851  |  0.3665357  |  0.4681134
## 0.5  |  2.398638  |  0.3648638  |  0.382857  |  0.4039488  |  0.6011956
## 0.6  |  2.782012  |  0.4022012  |  0.4210612  |  0.4431933  |  0.737774
## 0.7  |  3.203618  |  0.4413618  |  0.4611799  |  0.4844616  |  0.8761127
## 0.8  |  3.665375  |  0.4825375  |  0.5034144  |  0.5279667  |  1.014293
## 0.9  |  4.169402  |  0.5259402  |  0.5479872  |  0.5739437  |  1.150196
## 1  |  4.718041  |  0.5718041  |  0.5951443  |  0.6226526  |  1.281477
## 1.1  |  5.31388  |  0.620388  |  0.6451574  |  0.6743807  |  1.405548
## 1.2  |  5.95978  |  0.671978  |  0.6983269  |  0.7294456  |  1.519546
## 1.3  |  6.658902  |  0.7268902  |  0.7549847  |  0.7881981  |  1.620308
## 1.4  |  7.41474  |  0.785474  |  0.8154976  |  0.8510261  |  1.70434
## 1.5  |  8.231155  |  0.8481155  |  0.8802712  |  0.9183582  |  1.767776
## 1.6  |  9.112414  |  0.9152414  |  0.9497535  |  0.990668  |  1.80635
## 1.7  |  10.06323  |  0.9873235  |  1.02444  |  1.068479  |  1.81534
## 1.8  |  11.08883  |  1.064883  |  1.104877  |  1.15237  |  1.789534
## 1.9  |  12.19496  |  1.148496  |  1.19167  |  1.24298  |  1.723166
```

```
lines(e1$X,e1$Y, type='l', lwd=2, col='green',
     main="Euler",xlab="X", ylab="Y")
legend(0,15, # Posicion de la leyenda
       c('RK3','Euler'), # Nombre de lineas
       lty=c(1,1), # Pone los simbolos de prueba (lineas)
       lwd=c(1,2.5),col=c('red','green'))
```


RK3
RK3
Euler
y
x
0
5
10
15
0.0
0.5
1.0
1.5
2.0