

# Serie de Taylor

José Ramón Pérez Navarro

October 2019

## 0.1. Funcion seno(x) de polinomio de grado 1,3,5,7,9,11

### Código

```
program aproximacion_taylor
implicit none

!Declaracion de variables
REAL(kind=8),external:: senotaylor
REAL(kind=8)::y,x
INTEGER:: n,i,a

!Abrimos un open para obtener la grafica
open(11, file = 'salida.dat', status = 'unknown')
! polinomio grado 1, 3, 5, 7, 9 y 11

do n=1,11,2

! para el valor de la x yendo desde -100 hasta 100
do i=-100,100,1
x= 0.1 * i
y=senotaylor(x,n)
print*, x,y
write(11,*) x, y
end do
write(11,*) ''
end do
close (11)

end program aproximacion_taylor

!=====
```

```

function senotaylor(x,n)
!=====
implicit none
!declaracion de variables
real(kind=8),intent(in):: x
integer,intent(in):: n
real(kind=8):: senotaylor
real(kind=8)::a,b,c,d, s, t
integer:: j

s=0

! de 0 hasta n
do j=0,n,1 !taylor
a=(-1.0)**j
b=2*j+1
c=x**b
d=b

factorial:do
d=d-1
b=b*d
if(d==1) exit
if(d==0) then
b=1
exit
end if
end do factorial

!sumatoria

t=(a/b)*c
s=s+t
end do !taylor

senotaylor=s

```

```
end function senotaylor
```

Gráfica

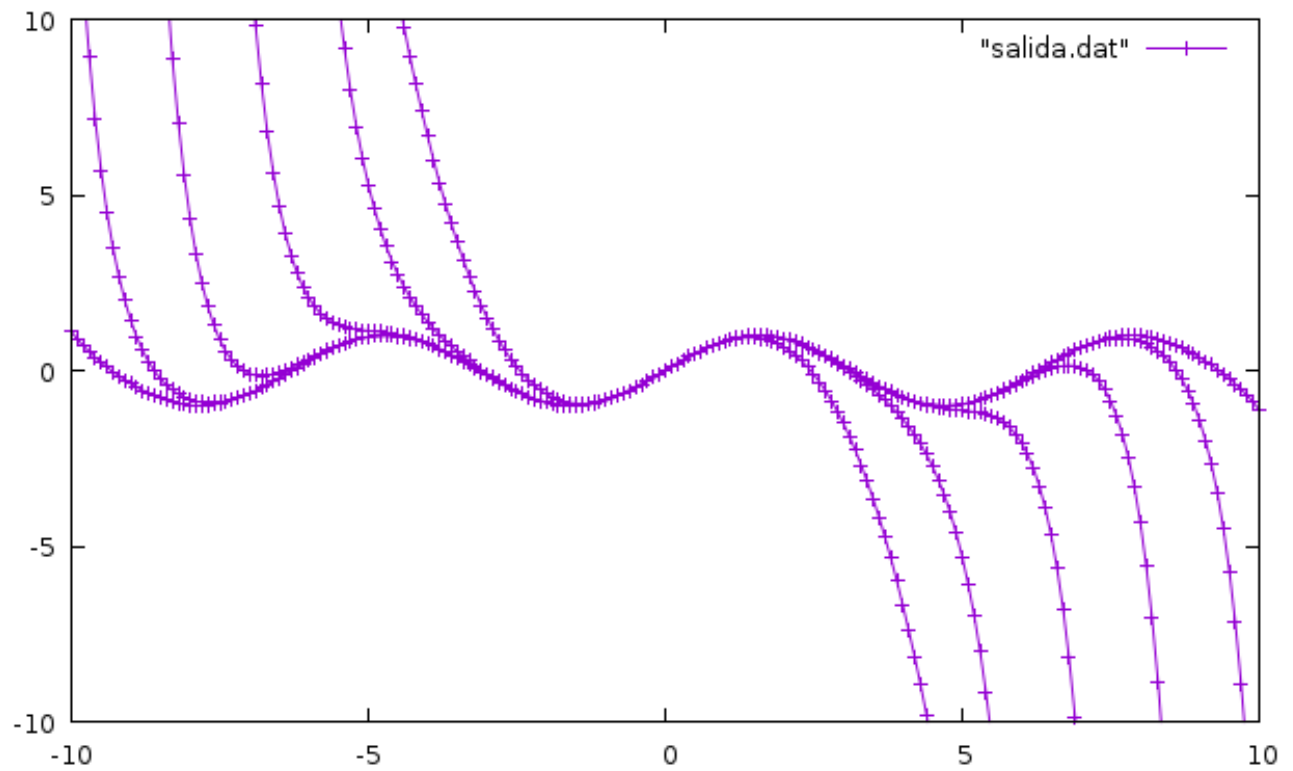


Figura 1:

## 0.2. Aproximación de la función $\ln(1+x)$

Código

```
program aproximacion_taylor
```

```

implicit none

!Declaracion de variables
real(kind=8),external:: ln_aprox
real(kind=8)::x,y
integer:: n,i

!Abrimos un documento para obtener la grafica
open(11,file='ln.dat')
!polinomio de grado 4 al 16
do n=4,16,1
  if (n>4 .and. n<7) cycle
  if (n>7 .and. n<11) cycle
  if (n>11 .and. n<16) cycle

  !para x de -250 a 250
  do i=-250,250,1
    x=0.01*i
    y=ln_aprox(x,n)
    print*,x,y
    write(11,*) x,y
  end do
  print*,' '
  write(11,*) ' '
end do
close (11)

end program aproximacion_taylor

!=====
function ln_aprox(x,n)
!=====
implicit none

!valores externos
real(kind=8),intent(in):: x
integer,intent(in):: n
real(kind=8):: ln_aprox

```

```

!variables
real(kind=8):: t,s,a,b,c
integer:: j

s=0
!grado 0 a n
do j=1,n,1
  a=(-1.0)**(j+1)
  b=x**j
  c=j

  t=a*(b/c)
  s=s+t
end do

ln_aprox=s

end function ln_aprox

```

**Gráfica**

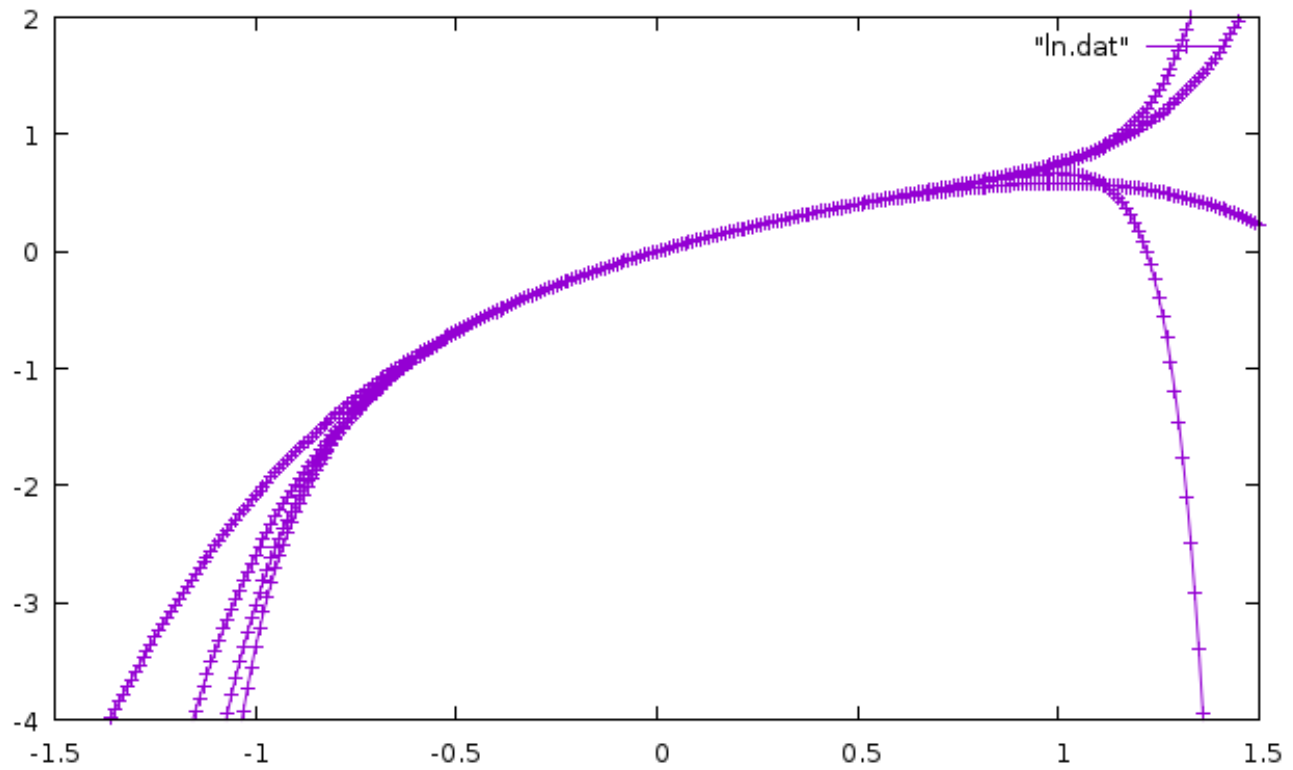


Figura 2:

### 0.3. Aproximación de la función exponencial

#### Código

```

program Aproximacion_taylor
implicit none

!Declaracion de variables necesarias
real(kind=8),external:: exp_aprox
real(kind=8)::y,x
integer:: n,i

open(11,file='exponencial.dat')
!hasta 8 terminos

```

```

do n=1,8,1

    ! para x
    do i=-500,500,1
        x=0.01 * i
        y=exp_aprox(x,n)
        print*,x,y
        write(11,*) x,y
    end do
    write(11,*) ' '
end do
close (11)

end program Aproximacion_taylor

!=====
function exp_aprox(x,n)
!=====
implicit none

!declaracion de variables
real(kind=8),intent(in):: x
integer,intent(in):: n
real(kind=8):: exp_aprox
real(kind=8):: s,t,c,b,a
integer:: j

s=0 ! suma

Aproximacion_taylor:do j=0,n,1
    b=x**j
    a=j
    c=a
    Factorial:do !calculamos el factorial
        if(c==0) then
            a=1
            exit

```



```

        end if
        c=c-1
        a=a*c
        if(c==1) exit
        if(c==0) then
            a=1
            exit
        end if
    end do Factorial

    t=b/a
    s=s+t
    end do Aproximacion_taylor

exp_aprox=s

end function exp_aprox

```

**Gráfica**

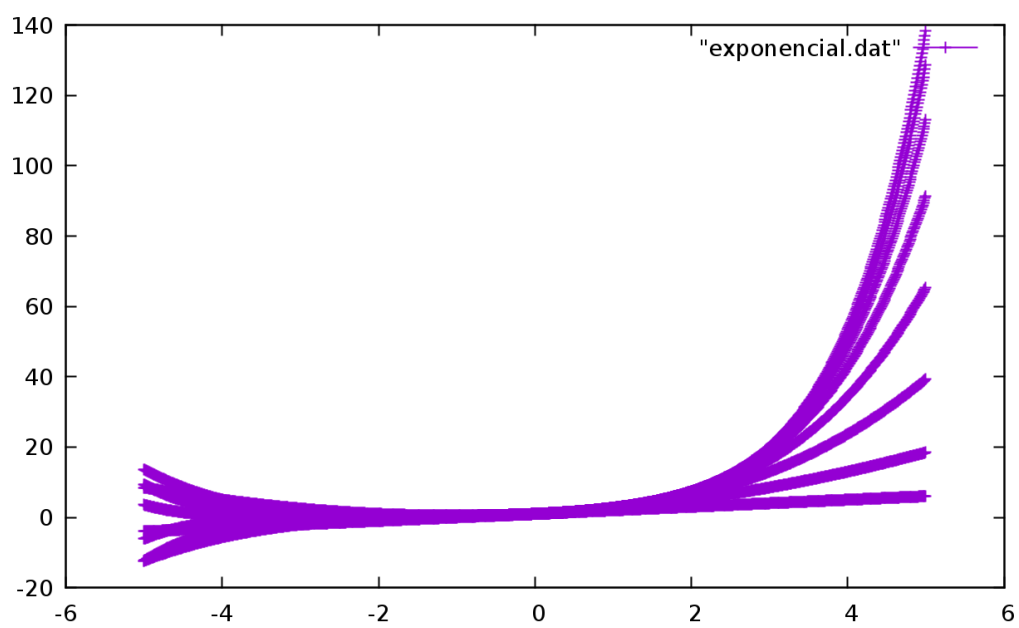


Figura 3: