Aproximacion de pade

José Ramón Pérez Navarro

October 2019

0.1. Problema

Aproximación del seno

```
program pade
implicit none
!declaracion de variables
real(kind=8), external :: seno_p
real(kind=8) :: seno_z, x, y, error, dx
real, parameter :: Pi= 3.1415926
integer :: i, n

!salida de datos

open (11, file = 'seno.dat')
  n =1000
  x = -2.0 * Pi
  dx= 4.0 * Pi/n
  do i = 1, n
    x= x + dx
     seno_z=Sin(x)
      y = seno_p(x)

      print*, x, seno_z, y !resultados
      write(11,*) x, seno_z, y
  end do

print*, ''


close (11)

end program pade

!==================

function seno_p(x)
```

```
!====================

implicit none

real (kind=8), intent(in):: x
real (kind=8) :: seno_p, seno_w, seno_v

seno_w = x - (x**3) * (2363.0/18183.0) + (x**5) * (12671.0/4363920.0)

seno_v=1 + (x**2) * (445.0/12122.0) + (x**4) * (601.0/872784.0) + (x**6)*(121.0/16

seno_p = seno_w/seno_v

end function seno_p
```

Gráfica

de pantalla de 2019-10-17 09-45-26.png

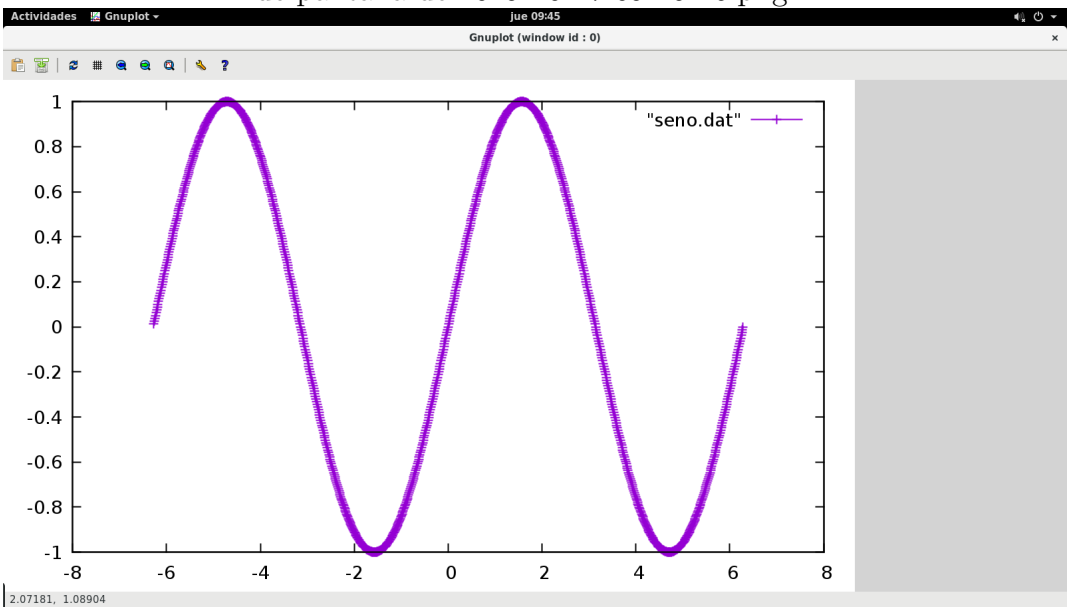


Figura 1:

Error relativo

```
program pade
implicit none

real(kind=8), external :: seno_p
real(kind=8) :: seno_z, x, y, error_p, dx
real, parameter :: Pi = 3.1415926
integer :: i, n

open (11, file = 'errorseno.dat')

  n = 1000
  x = -2.0 * Pi
  dx= 4.0 * Pi/n

   do i =1, n
       x = x + dx
        y = seno_p(x)
         seno_z=Sin(x)
          error_p = (seno_z - y) / seno_z
           print*, x, error_p
            write(11,*) x, error_p
   end do

   close (11)

end program pade

!==================

function seno_p(x)

!==================

implicit none

real (kind=8), intent(in):: x
```

```
real (kind=8) :: seno_p, seno_w, seno_v

seno_w = x - (x**3) * (2363.0/18183.0) + (x**5) * (12671.0/4363920.0)

seno_v = 1 + (x**2) * (445.0/12122.0) + (x**4) * (601.0/872784.0) + (x**6)*(121.0/

seno_p = seno_w/seno_v

end function seno_p
```

Gráfica del error relativo

de pantalla de 2019-10-17 10-21-48.png

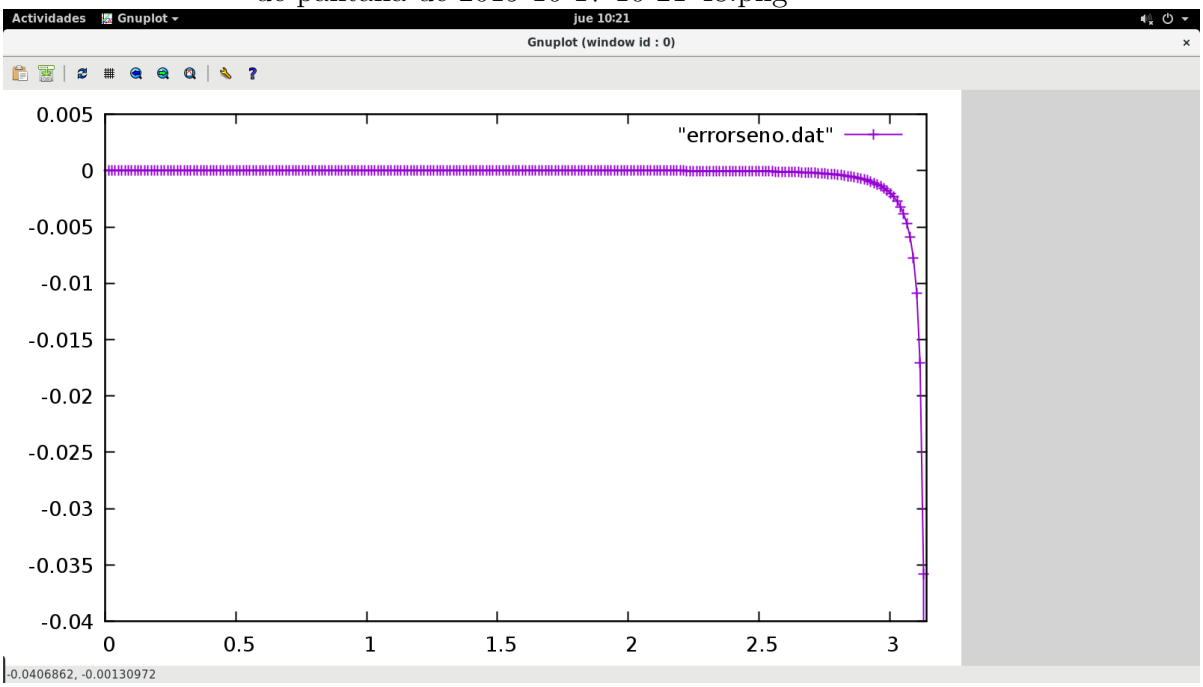


Figura 2:

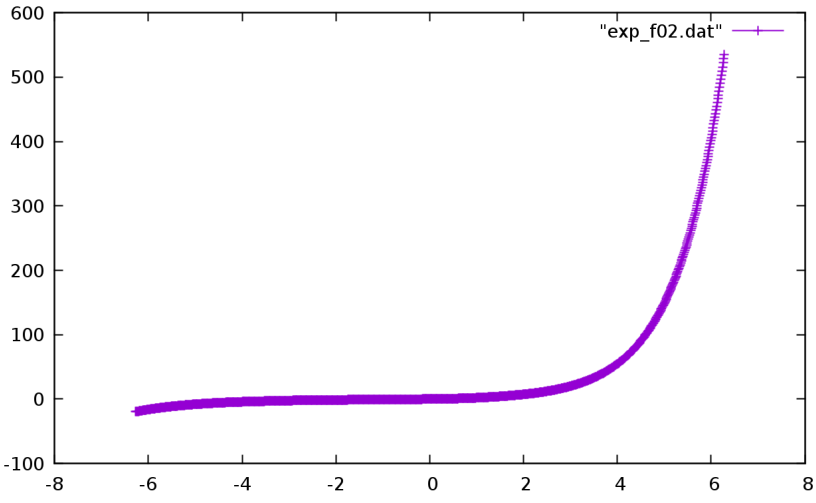
0.2. Problema

Funcion exponencial "F02"

```
  program pade
implicit none

!declaracion de variables
real(kind=8), external :: exp_f02
real(kind=8) :: x, y, error, exp_a, dx
real, parameter :: Pi= 3.1415926
integer :: i, n

!salida de datos

open (11, file = 'exp_f02.dat')

  n =1000
  x = -2.0 * Pi
  dx= 4.0 * Pi/n

  do i = 1, n
      x= x + dx
       y = exp_f02(x)
        exp_a= exp(x)
         error = exp_a -(y/exp_a)
          print*, x, error !resultados
           write(11,*) x, error
  end do

print*, ''

close (11)

end program pade

!==================
```

```
function exp_f02(x)

!====================

implicit none

real (kind=8), intent(in):: x
real (kind=8) :: exp_f02, w, v

w = 1.00

v = 1.00 - x + (x**2.00) * (1.00/2.00)

exp_f02 = w/v


end function exp_f02
```

Gráfica de función exponencial F02



exp$_f$02.*png*

Figura 3:

Funcion exponencial "F11"

```
program pade
implicit none

!declaracion de variables
real(kind=8), external :: exp_f11
real(kind=8) :: x, y, error, exp_a, dx
real, parameter :: Pi= 3.1415926
integer :: i, n

!salida de datos

open (11, file = 'exp_f11.dat')

  n = 1000
  x = -2.0 * Pi
  dx= 4.0 * Pi/n

  do i = 1, n
      x= x + dx
       y = exp_f11(x)
        exp_a= exp(x)
         error = exp_a -(y/exp_a)
          print*, x, error !resultados
           write(11,*) x, error
  end do

print*, ''

close (11)

end program pade

!===================

function exp_f11(x)
```

```
!====================

implicit none

real (kind=8), intent(in):: x
real (kind=8) :: exp_f11, w, v

w = 1.00 + x * (1.00/2.00)

v = 1.00 - x * (1.00/2.00)



exp_f11 = w/v

end function exp_f11
```

Gráfica de función exponencial F11

de pantalla de 2019-10-17 10-39-53.png

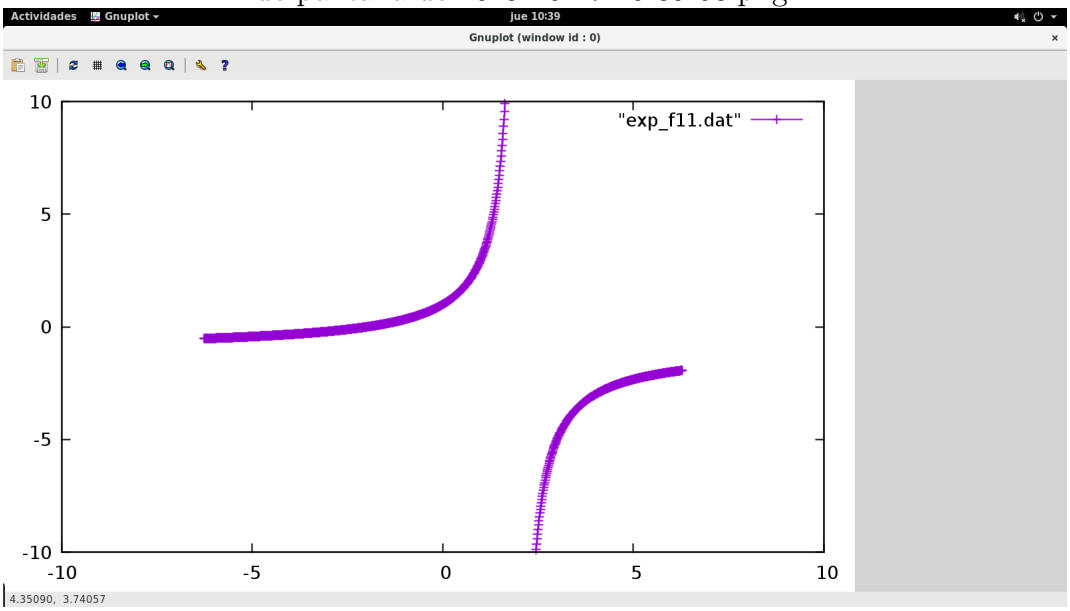


Figura 4:

Funcion exponencial "F20"

```
program pade
implicit none

!declaracion de variables
real(kind=8), external :: exp_f20
real(kind=8) :: x, y, error, exp_a, dx
real, parameter :: Pi = 3.1415926
integer :: i, n

!salida de datos

open (11, file = 'exp_f20.dat')
```

```
  n = 1000
  x = -2.0 * Pi
  dx= 4.0 * Pi/n

  do  i =1, n
       x = x + dx
        y = exp_f20(x)
         exp_a= exp(x)
          error = exp_a -(y/exp_a)
           print*, x, error, y!resultados
            write(11,*) x, error, y
  end do

print*, ''

close (11)

end program pade

!====================

function exp_f20(x)

!====================

implicit none

real (kind=8), intent(in):: x
real (kind=8) :: exp_f20, w, v

v = 1.00

w = 1.00 + x + (x**2.00) * (1.00/2.00)

exp_f20 = w/v

end function exp_f20
```

Función exponencial F20

(sin y).png

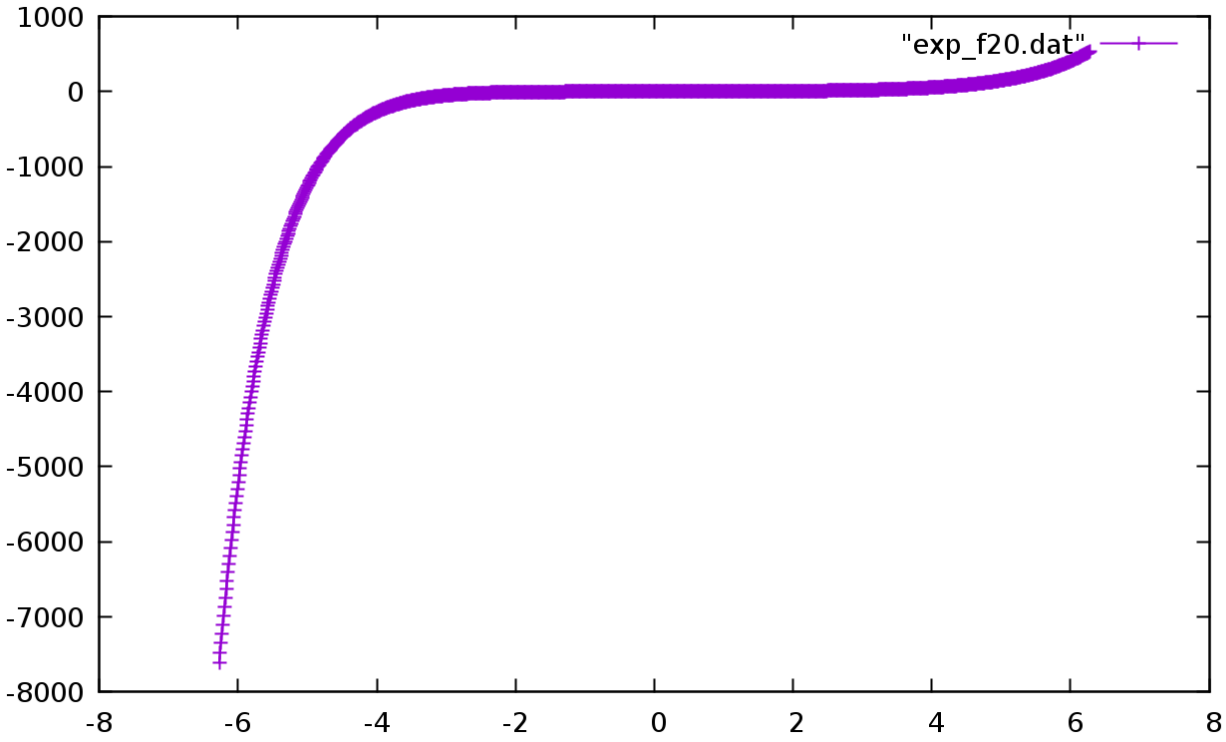


Figura 5: