EVALUACIÓN 1 - REPORTE

Problema 1

$$\sin(x) \approx \frac{(12671/4363920)x^5 - (2363/18183)x^3 + x}{1 + (445/12122)x^2 + (601/872784)x^4 + (121/16662240)x^6}$$

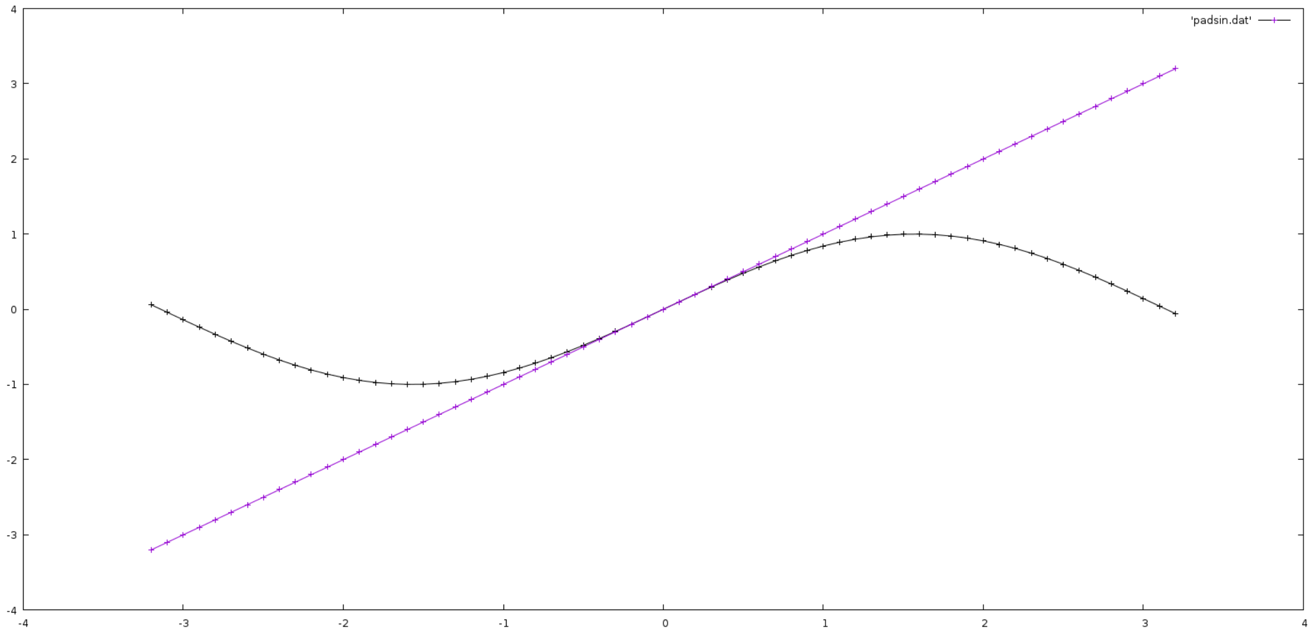
Construya un programa en Fortran que incluya una función externa $f_{56}(x)$, para el aproximante de Padé de la función sin(x), arriba expresada y poder generar los datos a un archivo para graficar con Gnuplot:

1. La función “sin(x)” y su representación de Padé “$f_{56}(x)$”, arriba expresada, en el rango ($-\pi, \pi$)

La sintaxis de Fortran para lograr este objetivo fue la siguiente

```fortran
program aprox_sin

    implicit none
    real:: x, sin_true, y, dt
    integer :: j, k
    real, external :: pade

 open (11, file = 'padsin.dat', status = 'unknown')
dt=0.1

    do j =-32,32
       x = float(j) * dt                      ! convert to a real
       sin_true = sin(x)
       write(11,*) x, sin_true, 0
       enddo
     write(11,*) " "

    do k = -32,32
         x = float(k) * dt
         y = pade(x)    !función definida abajo
         write(11,*) x, y, 1
    enddo
    write(11,*) " "
close (11)
end program aprox_sin
!===================================================================================================
function pade(x)  !necesita dos argumentos? como en exptaylor(x,n)
  implicit none
   ! define variables
   real, intent(in) :: x          !único input
   real :: pade                   !variable de salida

   ! variables locales
   real :: p

   !generando la aproximación
    p = ((12671/4363920 * (x**5)) - (2363/18183 * (x**3)) + x) & !numerador
    / (1 + (445/12122 * (x**2)) + (601/872784 * (x**4)) + (121/16662240 * (x**6))) !denominador

pade = p

end function pade
```

Lo cual produjo la siguiente gráfica en Gnuplot



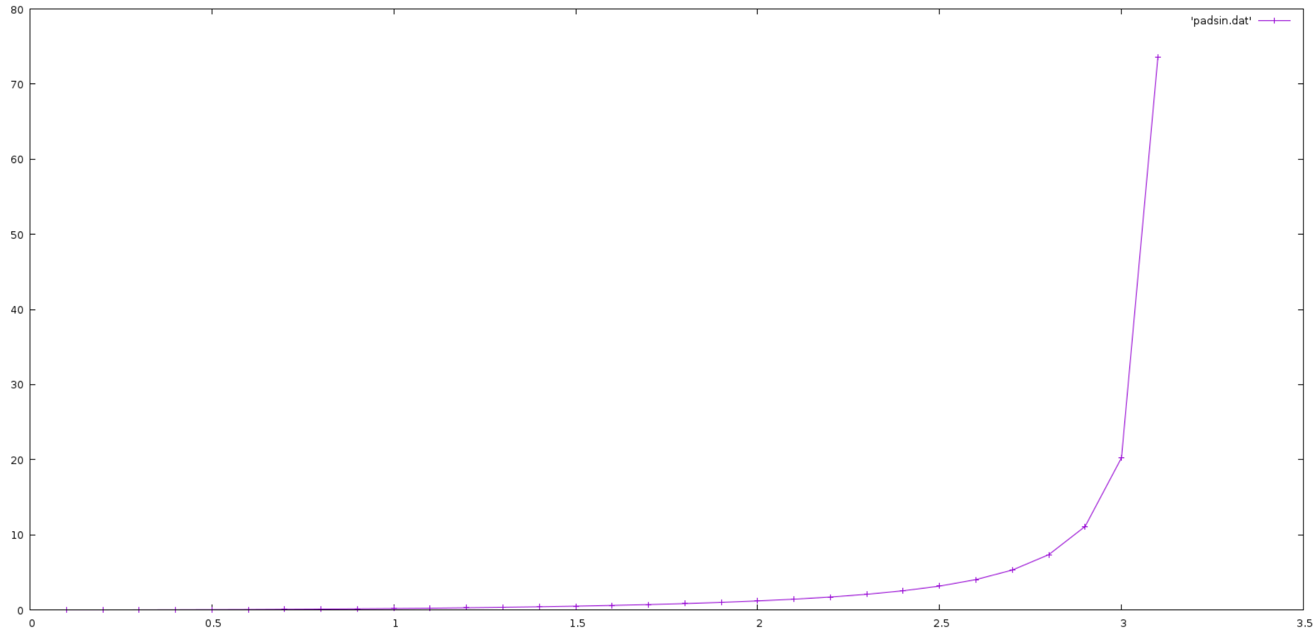
En negro se representa la función sin(x) original, mientras que en morado se representa la aproximación de Padé

2. Una gráfica del error relativo: Error Relativo = (sin(x) – f_{56}(x))/sin(x), para x en el intervalo (0, π).

Para ello, se redactó lo siguiente

```
program aprox_sin

    implicit none
    real:: x, sin_true, y, dt, Er
    integer :: l
    real, external :: pade

 open (11, file = 'padsin.dat', status = 'unknown')
dt=0.1

    do l = 0,32
         x = float(l) * dt
         sin_true = sin(x)
         y = pade(x)   !función definida abajo
         Er = abs(y - sin_true) / sin_true
         write(11,*) x, Er
    enddo

close (11)
end program aprox_sin
!======================================================================================================
function pade(x)  !necesita dos argumentos? como en exptaylor(x,n)
  implicit none
   ! define variables
   real, intent(in) :: x         !único input
   real :: pade                  !variable de salida

   ! variables locales
   real :: p

   !generando la aproximación
    p = ((12671/4363920 * (x**5)) - (2363/18183 * (x**3)) + x) & !numerador
    / (1 + (445/12122 * (x**2)) + (601/872784 * (x**4)) + (121/16662240 * (x**6))) !denominador

pade = p

end function pade
```

Y en la gráfica se mostró lo siguiente, se hizo una corrección de un último dato erróneo.



Problema 2

Construya un programa en Fortran, que se apoye en funciones externas para calcular el Error Relativo de utilizar un aproximante de Padé $f_{mn}(x)$, para la función exponencial exp(z).

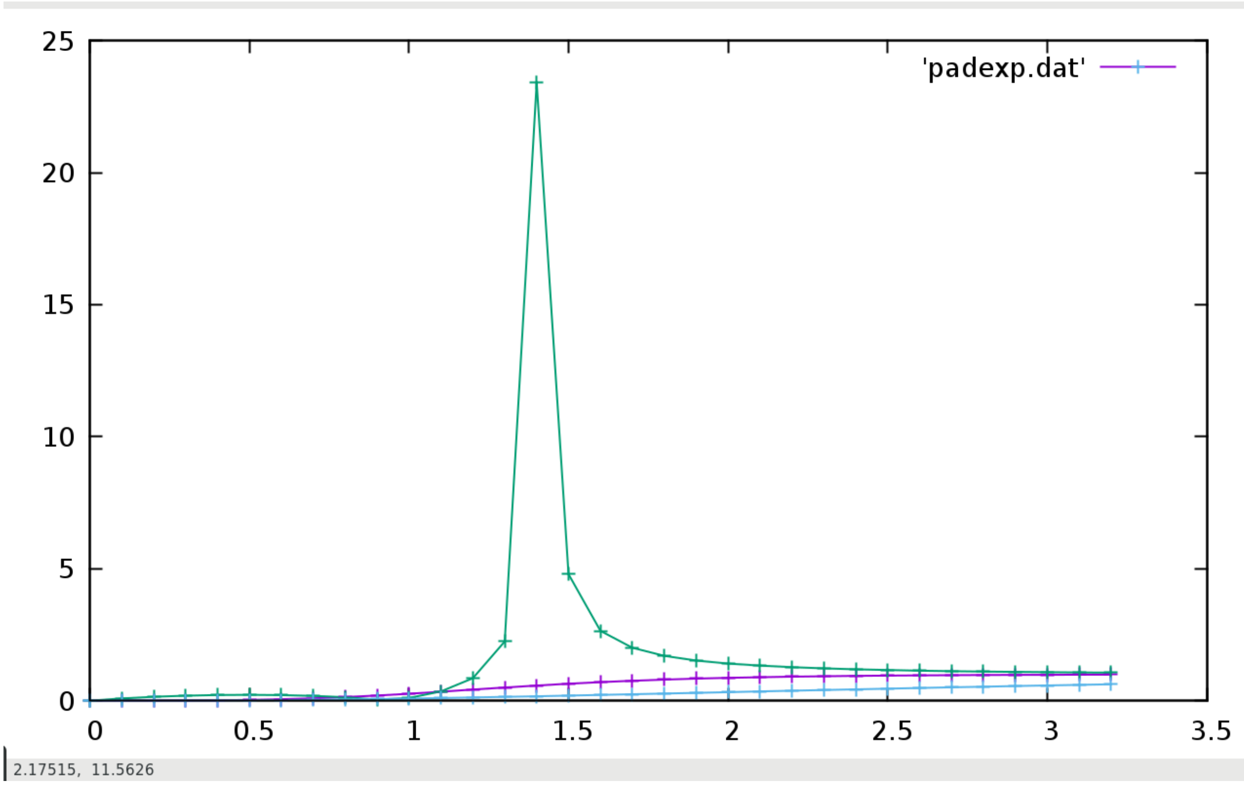
Sintaxis:

```
program aprox_sin

    implicit none
    real:: x, exp_true, y, dt, Er
    integer :: l
    real, external :: pade02, pade11, pade20

 open (11, file = 'padexp.dat', status = 'unknown')
dt=0.1

    do l = 0,32
         x = float(l) * dt
         exp_true = exp(x)
         y = pade02(x)   !función definida abajo
         Er = abs(y - exp_true) / exp_true
         write(11,*) x, Er, 1
    enddo
write(11,*) " "
    do l = 0,32
         x = float(l) * dt
         exp_true = exp(x)
         y = pade11(x)   !función definida abajo
         Er = abs(y - exp_true) / exp_true
         write(11,*) x, Er, 2
    enddo
write(11,*) " "
 do l = 0,32
         x = float(l) * dt
         exp_true = exp(x)
         y = pade20(x)   !función definida abajo
         Er = abs(y - exp_true) / exp_true
         write(11,*) x, Er, 3
    enddo
close (11)
end program aprox_sin
!====================================================================================================
function pade02(x)
  implicit none
   ! define variables
   real, intent(in) :: x         !único input
   real :: pade02                  !variable de salida

   ! variables locales
   real :: p
```


```
    ! variables locales
    real :: p
```


```
    !generando la aproximación
     p = 1/(1 - x + (0.5 * (x**2)))

 pade02 = p

 end function pade02

 !==============================================================================
 function pade11(x)
   implicit none
    ! define variables
    real, intent(in) :: x         !único input
    real :: pade11                  !variable de salida

    ! variables locales
    real :: p

    !generando la aproximación
      p = (1 + (0.5 * (x**2)))/(1 - (0.5 * (x**2)))

 pade11 = p

 end function pade11

 !==============================================================================
 function pade20(x)
   implicit none
    ! define variables
    real, intent(in) :: x         !único input
    real :: pade20                  !variable de salida

    ! variables locales
    real :: p

    !generando la aproximación
     p = (1 + x + (0.5 * (x**2)))

 pade20 = p

 end function pade20
```

Gráfica:



- La línea morada representa la f02(x)
- La línea verde representa la f11(x), siendo esta la menos acertada, pues tiene mucho error en x=1.5 aproximadamente.
- La línea azul representa la f20, siendo esta la que tiene menos error, aunque es muy comparable a f02(x)