Examen de Prácticas de Laboratorio de Cálculo

Grupo A: PL-01

Resolver los siguientes ejercicios utilizando `Matlab` *indicando* ***los comandos utilizados para resolverlos****,* ***las respuestas obtenidas*** *y* ***las representaciones gráficas que se piden****.*
Adjuntar un archivo PDF en la tarea del Campus Virtual creada para esta prueba, identificándolo con vuestro nombre y apellidos.

Corrección

1. **Dada la función:**

$$f(x) = \frac{1}{1+x^2}$$

a) **(1 Punto) Calcular los límites:** $\lim_{x\to+\infty} f(x)$ **y** $\lim_{x\to-\infty} f(x)$**.**

Se introduce la función en `Matlab`, se muestra por pantalla utilizando el comando `pretty` (para comprobar que se ha informado a `Matlab` de la función correcta, ¡no faltan partén-tesis!) y se ejecutan los comandos para calcular los límites pedidos:

```
syms x
f(x)=1/(1+x^2);
pretty(f)
limit(f(x),x,+Inf)
limit(f(x),x,-Inf)
```

Se obtiene:

```
    1
------
 2
x  + 1


ans =

0


ans =

0
```

Esto es, el valor de cada uno de los límites es 0.

b) **(1 Punto) Obtener los intervalos de monotonía (en dónde es creciente y decreciente), proporcionando los máximos y mínimos que tuviere (sean relativos o absolutos).**

Se calcula la derivada (primera) de la función y los números reales en los que se anula:

```
fprima(x)=diff(f,x);
double(solve(fprima))
```

```
ans =

     0
```

El dominio de la función es $\mathbb{R}$, puesto que el denominador que aparece en la definición de la función no se anula en ningún número real ($x^2 + 1 \geq 1$), por tanto, no se debe añadir ningún otro *valor conflictivo* para estudiar la monotonía de la función. Evaluamos la función en cada una de las regiones de los números reales que divide 0; se toman (por ejemplo) los valores -1 y 1.

```
fprima([-1,1])
```

```
ans =

[1/2, -1/2]
```

Por todo lo obtenido, se deduce que la función es creciente ($f'(x) \geq 0$) sobre el intervalo $(-\infty, 0)$ y decreciente ($f'(x) \leq 0$) sobre el intervalo $(0, +\infty)$. Así, la función posee un máximo en $x = 0$.

c) **(1 Punto) Obtener los intervalos de curvatura (en dónde es cóncava y convexa), proporcionando los puntos de inflexión que tuviere.**

Se calcula la derivada segunda de la función y los números reales en los que se anula:

```
fsegunda(x)=diff(f,x,2);
double(solve(fsegunda))
```

```
ans =

   -0.5774
    0.5774
```

Se calculan las imágenes de la función en cada una de las tres regiones delimitadas por los valores obtenidos (por ejemplo, en -2, 0 y 2).

```
fsegunda([-2,0,2])
```

```
ans =

[22/125, -2, 22/125]
```

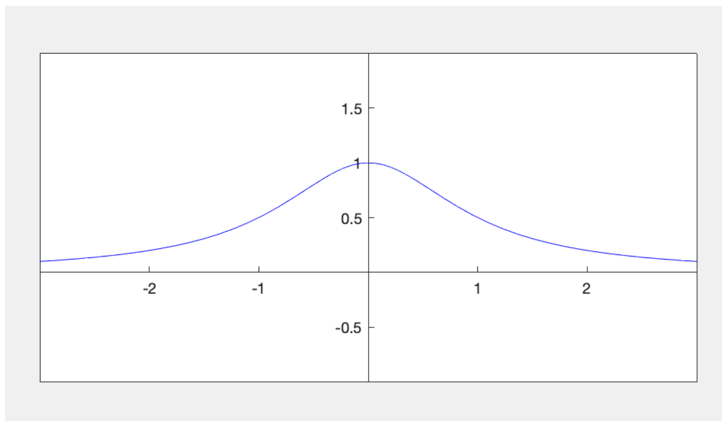
La función es cóncava (convexa para Set) en los intervalos $(-\infty, -0.5774) \cup (0.5774, +\infty)$ y convexa (cóncava para Set) en el intervalo $(-0.5774, 0.5774)$. Por tanto, la función posee dos puntos de inflexión situados en $x = -0.5774$ y $x = 0.5774$.

d) **(1 Punto) Representar la función sobre el intervalo $[-3, 3]$ en color azul y comprobar los resultados obtenidos analíticamente.**

Se ejecuta el comando `fplot` con los parámetros apropiados para representarla sobre el intervalo de abscisas exigido por el enunciado y en color azul. Además, se colocan los ejes de coordenadas de manera que se corten en el origen $(0, 0)$, se igualan las escalas de ambos ejes y, con el objetivo de visualizar la representación gráfica correctamente, se define también el rango de las imágenes en el intervalo $[-1, 2]$:

```
fplot(f,[-3,3],'b')
ax=gca;
ax.XAxisLocation='Origin';
ax.YAxisLocation='Origin';
axis equal
axis([-3,3,-1,2])
```

Se muestra el resultado obtenido:



Todo cuadra con los resultados obtenidos en los apartados anteriores.

2. *a*) **(1 Punto) Calcular el *Polinomio de Taylor* de grado 2 asociado a la función alrededor del punto $x = 0$.**

Se pide a `Matlab` el *Polinomio de Taylor* con los parámetros adecuados para hacerlo: alrededor del punto $x = 0$ y con grado $n = 2$, en general, tendría tres sumandos (de ahí 'order', 3), aunque en este caso particular sólo aparecen dos, como se muestra en la salida. La salida se muestra con el comando `pretty`.

```
p2(x)=taylor(f,x,0,'order',3);
pretty(p2)
```

```
     2
1 - x
```

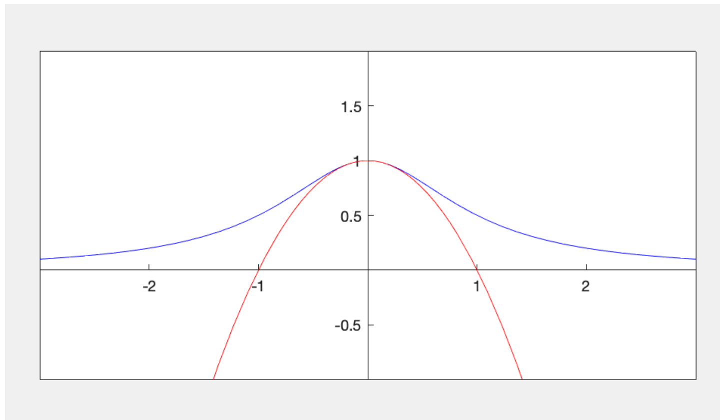
Esto es, el *Polnomio de Taylor* pedido es $p_2(x) = 1 - x^2$.

b) **(1 Punto) Representar el *Polinomio de Taylor* de grado 2 junto con la función sobre el intervalo $[-3, 3]$ en color rojo.**

Se mantiene la representación de la función (lo exige el ejercicio) y se representa el polinomio en color rojo.

```
hold on
fplot(p2,[-3,3],'r')
```

Véase la figura que se obtiene:



c) **(1 Punto) Utilizar el polinomio anterior para aproximar el valor que toma la función en $x = 0.6$ y calcular el error cometido con la aproximación anterior. Expresar ambos valores numéricamente (no simbólicamente).**

Se realizan los cálculos directamente y se muestran los resultados.

```
aprox1=double(p2(0.6))
error1=double(f(0.6)-p2(0.6))


aprox1 =

    0.6400


error1 =

    0.0953
```

d) **(0.5 Puntos) Calcular el *Polinomio de Taylor* de grado 4 asociado a la función alrededor del punto $x = 1$ y utilizarlo para aproximar el valor que toma la función en $x = 0.6$, calculando el error y expresándolo numéricamente.**

Se pide a `Matlab` el *Polinomio de Taylor* con los parámetros adecuados para hacerlo alrededor del punto $x = 1$ y con grado $n = 4$. La salida se muestra con el comando `pretty`.

```
p4(x)=taylor(f,x,1,'order',5);
pretty(p4)
       2                4
 (x - 1)    x    (x - 1)
-------- - - - -------- + 1
    4       2       8
```

Puede obtenerse la expresión tradicional del polinomio añadiendo el comando `expand`.

```
expand(p4)
```

Para obtener:

```
- x^4/8 + x^3/2 - x^2/2 - x/2 + 9/8
```

Es decir, el *Polinomio de Taylor* de grado $n = 4$ alrededor de $x = 1$ es:

$$p_4(x) = -\frac{1}{8}x^4 + \frac{1}{2}x^3 - \frac{1}{2}x^2 - \frac{1}{2}x + \frac{9}{8}$$

Finalmente, se obtienen la aproximación y el error pedidos:

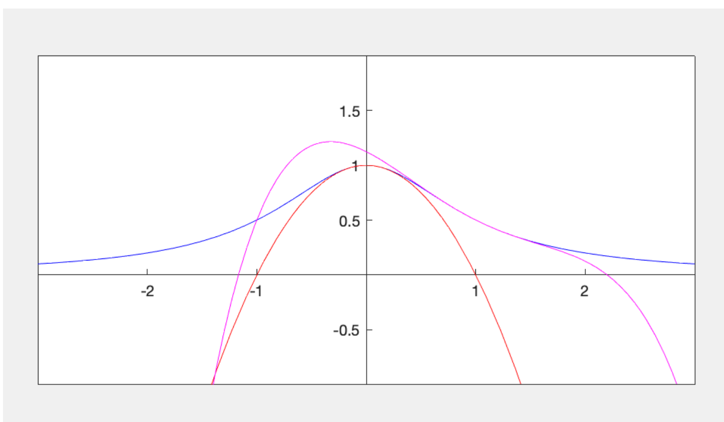
```
aprox2=double(p4(0.6))
error2=double(f(0.6)-p4(0.6))
```

```
aprox2 =

    0.7368


error2 =

   -0.0015
```

e) **(0.5 Puntos) Representar el *Polinomio de Taylor* del apartado anterior junto con la función y el *Polinomio de Taylor* de grado 2.**

Se mantiene lo representado y añadimos el nuevo polinomio (en color magenta).

```
fplot(p4,[-3,3],'m')
```



3. **(1 Punto) Calcular el área que encierra la función sobre el eje de abscisas.**

Para calcular el área que encierra la función sobre el eje de abscisas $(-\infty, +\infty)$, sólo es necesario el siguiente comando:

```
area1=int(f,x,-Inf,+Inf)
```

Y se obtiene:

```
area1 =

pi
```

4. **(1 Punto) Calcular el área que encierra el *Polinomio de Taylor* de grado 4 calculado en el ejercicio anterior sobre el eje de abscisas.**

   Deben calcularse los puntos de corte del *Polinomio de Taylor* de grado $n = 4$ calculado en el apartado 2.c) con el eje de abscisas.

```
double(solve(p4))
```

```
ans =

  -1.1673 + 0.0000i
   1.4831 - 1.1416i
   1.4831 + 1.1416i
   2.2010 + 0.0000i
```

   Teniendo en cuenta que las soluciones complejas no tienen interés para el problema en cuestión, los cortes con el eje de abscisas se producen en $x = -1.1673$ y en $x = 2.2010$ (coherente con la representación gráfica). Estos valores se convierten en los límites de integración del último comando que debe introducirse para finalizar el examen.

```
area2=double(int(p4,x,-1.1673,2.2010))
```

```
area2 =

    2.2326
```