

Sistemas de Cálculo Simbólico

Trabajo final

Rubén Darias Bello – Luciano Rubio Romero
Grupo 0416, Curso 2007/08

Índice

9.7 Extremos Condicionados

19.5 Filtrado

22.4 Método de Kasiski

23.5 Función seno cardinal

Extremos condicionados

Ruben Darias Bello - Luciano Rubio Romero
Sistemas de Cálculo Simbólico
Grupo 0416, Curso 2007/08

▼ Enunciado

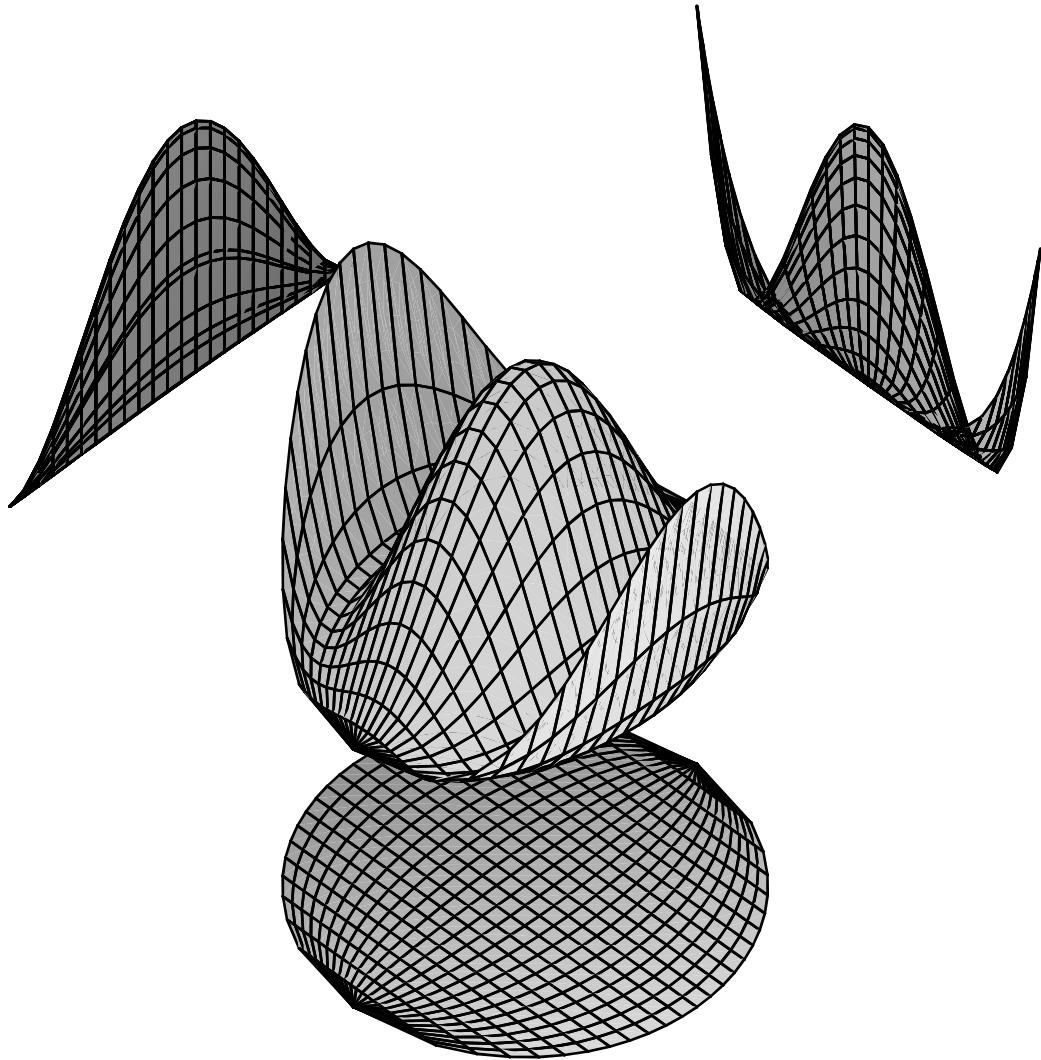
Dada la función $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ tal que $f(x,y) = (x^2 + 2y^2 - 1)^2$, se pide:

▼ Apartado a

Dibujar la función f en el dominio $D = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$ y sus proyecciones sobre los planos coordenados $x = 0$, $y = 0$ y $z = 0$.

▼ Solución

```
> with(plots):
> with(plottools):
> Funcion := (x,y) → (x² + 2·y² - 1)²
          Funcion := (x,y) → (x² + 2 y² - 1)²
                                         (2.1.1)
> Plotfuncion := plot3d(Funcion(x,y), x = -1..1, y = -sqrt(1-x²)..sqrt(1-x²)):
> Prox := project(Plotfuncion, [ [-2, 0, 0], [ -2, 1, 0], [ -2, 0, 1] ]):
> Proy := project(Plotfuncion, [ [0, -2, 0], [1, -2, 0], [0, -2, 1] ]):
> Proz := project(Plotfuncion, [ [0, 0, -1], [1, 0, -1], [0, 1, -1] ]):
> display([Plotfuncion, Prox, Proy, Proz]);
```



Apartado b

Hallar los puntos estacionarios de f en \mathbb{R}^2 y determinar, si procede, el tipo de extremos que se alcanzan.

Solución

Hallamos el jacobiano de la función para ver en que puntos se anula y así conocer los puntos estacionarios. Seguidamente, miramos la función para conocer de qué tipo son los extremos.

```
> with(linalg):
> Jacobiano:=jacobian([Funcion(x, y)], [x, y]);
Jacobian := [4 (x^2 + 2 y^2 - 1) x  8 (x^2 + 2 y^2 - 1) y]           (3.1.1)
> solve({{Jacobian[1, 1]}=0, {Jacobian[1, 2]}=0});
{y = 0, x = 0}, {y = RootOf(-1
+ 2 _Z^2, label = _L1), x = 0}, {x = 1, y = 0}, {y = 0, x = -1}, {y = y,
x = RootOf(_Z^2 - 1 + 2 y^2)}           (3.1.2)
```

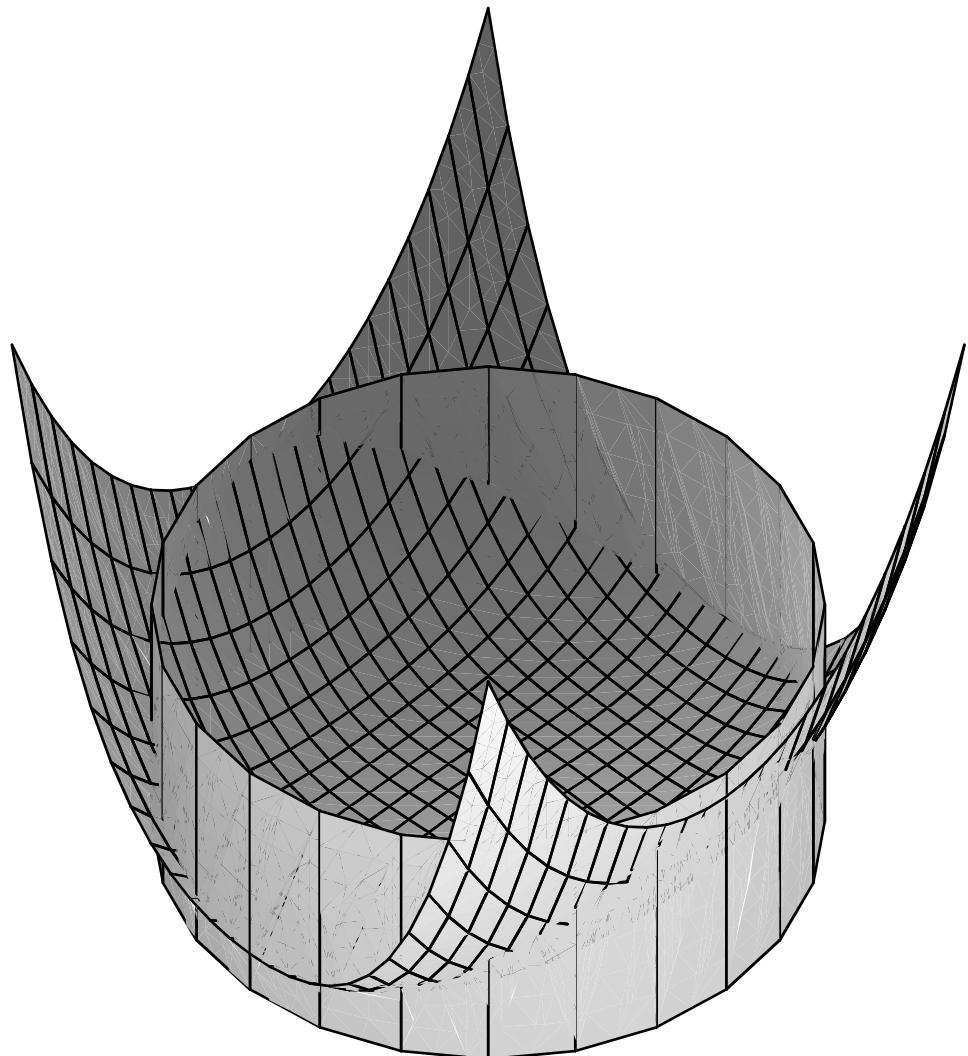
Viendo el plot de la función, nos damos cuenta de que los puntos estacionarios de $(1, 0)$ y $(-1, 0)$ son mínimos absolutos y el punto $(0, 0)$ es un máximo local.

Apartado c

Dibujar conjuntamente la curva, C, intersección de las superficies $z = f(x, y)$ y $x^2 + y^2 - 4 = 0$, y sus proyecciones sobre los planos coordinados.

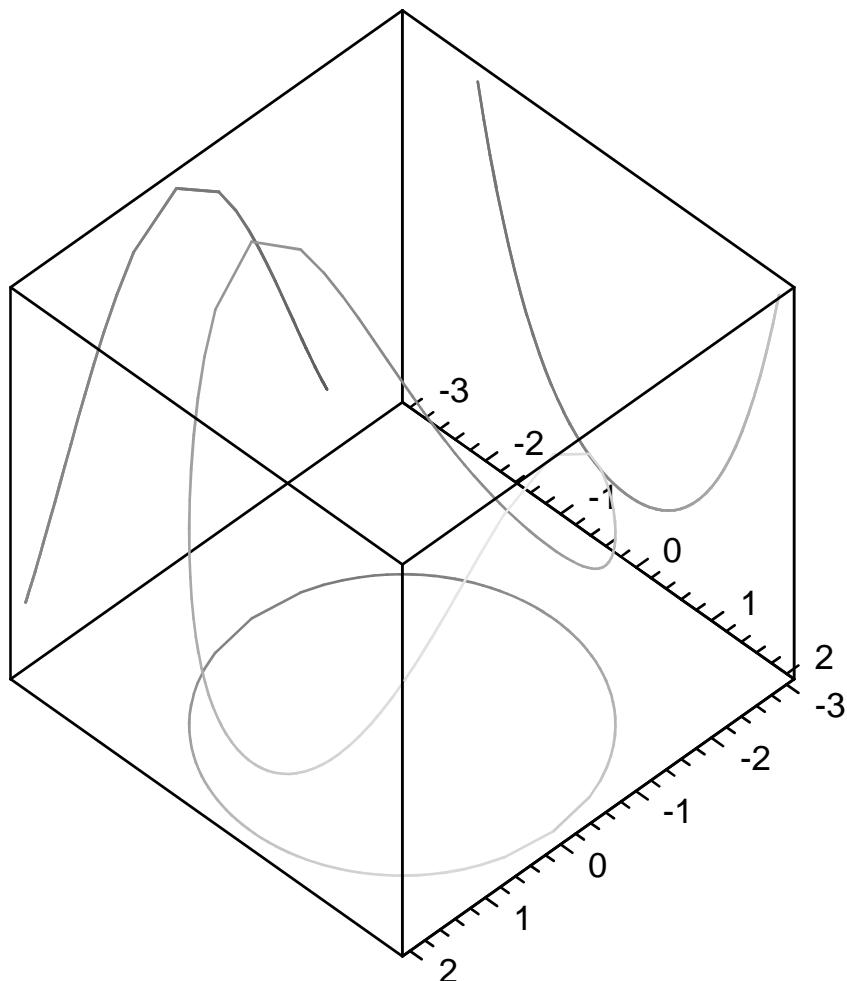
Solución

```
> Plot1 := plot3d(Funcion(x, y), x = -2 .. 2, y = -2 .. 2) :  
> Cilindro := cylinder([0, 0, 0], 2, 55, capped=false):  
> display([Plot1, Cilindro]);
```



Sabiendo que $x^2 + y^2 - 4 = 0$ y que $(x^2 + 2y^2 - 1)^2 = z$ y haciendo un simple sistema de ecuaciones, sacamos que la curva parametrizada en $y=t$ tiene que ser de la forma que se explica más abajo.

```
> Curva := spacecurve([sqrt(-t^2 + 4), t, (t^2 + 3)^2], [-sqrt(-t^2 + 4), t, (t^2 + 3)^2], t =  
-2 .. 2);  
> Cx := project(Curva, [-3, 0, 0], [-3, 1, 0], [-3, 0, 1]);  
> Cy := project(Curva, [0, -3, 0], [1, -3, 0], [0, -3, 1]);  
> Cz := project(Curva, [0, 0, 0], [1, 0, 0], [0, 1, 0]);  
> display([Curva, Cx, Cy, Cz], axes = boxed);
```



Apartado d

Calcular la longitud de C.

Solución

Usando el comando Lineint que nos calcula la integral de la línea seguida por la parametrización en $y=t$, obtenemos la forma inerte y una aproximación del resultado final.

$$> Longitud := \text{student}[Lineint]\left(1, x = \sqrt{4 - t^2}, y = t, z = (3 + t^2)^2, t = -2 .. 2\right);$$

$$Longitud := \int_{-2}^2 \sqrt{\left(\frac{dx}{dt} t\right)^2 + \left(\frac{dy}{dt} ((t^2 + 3)^2)\right)^2 + \left(\frac{dz}{dt} (\sqrt{-t^2 + 4})\right)^2} dt \quad (5.1.1)$$

$$> \text{value}(Longitud);$$

$$\int_{-2}^2 \sqrt{1 + 16(t^2 + 3)^2 t^2 + \frac{t^2}{-t^2 + 4}} dt \quad (5.1.2)$$

$$> \text{evalf}(\%);$$

$$80.53062494 \quad (5.1.3)$$

Apartado e

Justificar la existencia de extremos de f en la bola cerrada $B_2(0, 0)$ y determinarlos por los siguientes métodos:

▼ Solución

Multiplicadores de Lagrange

Tenemos que nuestra bola cerrada no es más que una esfera centrada en cero y de radio 2. Con lo cual y teniendo su ecuación, hallamos máximos y mínimos dentro de la bola usando el sistema de Lagrange.

```
> Auxiliar := x^2 + y^2 - 4 :  
> Func := Funcion(x, y) + a · Auxiliar;  
          Func := (x^2 + 2 y^2 - 1)^2 + a (x^2 + y^2 - 4)           (6.1.1)  
> solve( {diff(Func, x) = 0, diff(Func, y) = 0, Auxiliar = 0} );  
{y = 2, a = -28, x = 0}, {y = -2, a = -28, x = 0}, {a = -6, x = 2, y = 0}, {      (6.1.2)  
a = -6, x = -2, y = 0}, {x = RootOf(_Z^2 - 7), a = 0, y = RootOf(3 + _Z^2)}
```

▼ Apartado f

Interpretar los resultados obtenidos al aplicar el comando extrema.

▼ Solución

```
> extrema(Funcion(x, y), Auxiliar, {x, y, z});  
          {0, 49}           (7.1.1)
```

El comando extrema utiliza el método de multiplicadores de Lagrange para sacar los máximos y mínimos dentro de un límite marcado.

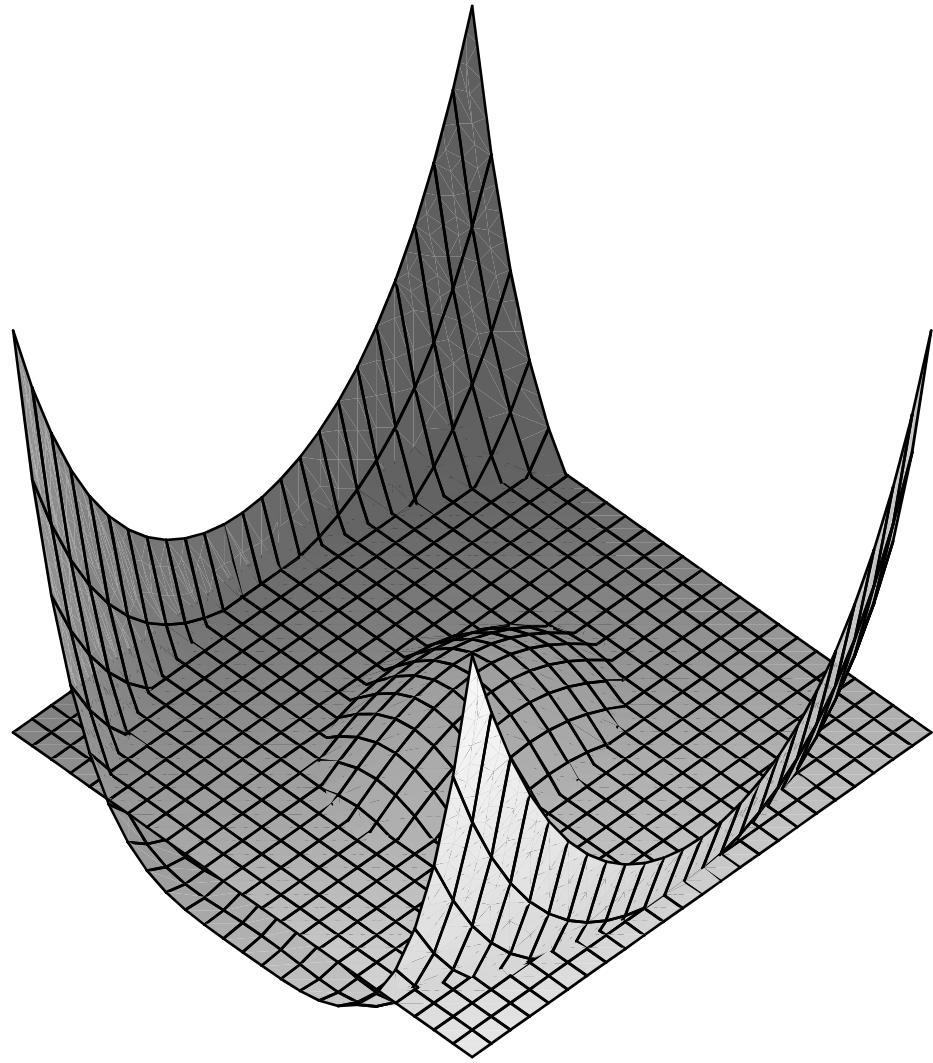
▼ Apartado g

Calcular el volumen de las regiones cerradas limitadas por la superficie $z=f(x, y)$ y el plano $z=\frac{1}{2}$.

▼ Solución

Pintamos la función junto a la superficie para saber qué tenemos que integrar.

```
> A := plot3d(Funcion(x, y), x = -1 .. 1, y = -1 .. 1) :  
> B := plot3d(1/2, x = -1 .. 1, y = -1 .. 1) :  
> display( {A, B});
```



A la ecuación inicial le restamos $\frac{1}{2}$ e integramos entre los límites que vemos en la figura.

$$\begin{aligned}
 > \text{student}[Tripleint]\left(\left(Funcion(x,y)-\frac{1}{2}\right),y\right. \\
 &= -\sqrt{-\frac{\sqrt{-z}+x^2+1}{2}}.\sqrt{\frac{-\sqrt{-z}+x^2+1}{2}}, z=\frac{1}{2}..1, x=\frac{-1}{2}..\frac{1}{2}\Big); \\
 &\int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}}^1 \int_{-\frac{1}{2}\sqrt{-2\sqrt{-z}+2x^2+2}}^{\frac{1}{2}\sqrt{-2\sqrt{-z}+2x^2+2}} (x^2+2y^2-1)^2 - \frac{1}{2} dy dz dx \quad (8.1.1)
 \end{aligned}$$

$$\begin{aligned}
 > \text{value}(\%); \\
 & -\frac{649}{100800}\sqrt{2} - \frac{19}{280}\sqrt{5-2\sqrt{2}} - \frac{7}{40}\ln(-2 \\
 & + \frac{1}{2}\sqrt{2}\sqrt{12+2\sqrt{2}}\sqrt{4-2\sqrt{2}}) + \frac{443}{7200}\sqrt{2}\sqrt{5-2\sqrt{2}} \\
 & + \frac{17}{160}\sqrt{2}\ln\left(-2+\frac{1}{2}\sqrt{2}\sqrt{12+2\sqrt{2}}\sqrt{4-2\sqrt{2}}\right) - \frac{17}{160}\sqrt{2}\ln(2
 \end{aligned} \quad (8.1.2)$$

$$\begin{aligned}
& \left[\left[+ \frac{1}{2} \sqrt{2} \sqrt{12 + 2\sqrt{2}} \sqrt{4 - 2\sqrt{2}} \right] + \frac{7}{40} \ln\left(2 + \frac{1}{2} \sqrt{2} \sqrt{12 + 2\sqrt{2}} \sqrt{4 - 2\sqrt{2}}\right) \right] \\
& > \text{evalf}(\%); \quad 0.0600192458 \tag{8.1.3}
\end{aligned}$$

Filtrado

Ruben Darias Bello - Luciano Rubio Romero
Sistemas de Cálculo Simbólico
Grupo 0416, Curso 2007/08

▼ Enunciado

Dada la señal $x(t) = \sum_{n=1}^5 \cos(4(n+1)\pi t)$ y la sucesión de funciones (f_n) , en donde

$$f_n(t) = \frac{1}{1 + (\epsilon t)^{2n}} \text{ con } t \in \mathbb{R} \text{ y } \epsilon \in [0, 1],$$

se pide:

▼ Apartado (a)

Generar, utilizando exclusivamente el comando seq, la lista de las imágenes $f_5(t)$ para los valores $\epsilon = 0, 0.2, 0.4 \dots 1$.

▼ Solución

Definimos la sucesión de funciones:

$$> f := (t, n, \epsilon) \rightarrow \frac{1}{1 + (\epsilon \cdot t)^{2n}} :$$

Y generamos las imágenes $f_5(t)$ únicamente con seq:

$$\begin{aligned} &> \text{imágenes} := [\text{seq}(f(t, 5, 0.2 \cdot i), i = 0 .. 5)]; \\ &\text{imágenes} := \left[1., \frac{1}{1 + 1.024 \cdot 10^{-7} t^{10}}, \frac{1}{1 + 0.0001048576 t^{10}}, \frac{1}{1 + 0.0060466176 t^{10}}, \right. \\ &\quad \left. \frac{1}{1 + 0.1073741824 t^{10}}, \frac{1}{1 + 1.000000000 t^{10}} \right] \end{aligned} \quad (2.1.1)$$

▼ Apartado (b)

Representar en un mosaico 3×2 correspondiente a los valores $\epsilon = 0.5, 0.6, \dots, 1$ conteniendo cada elemento las gráficas de las funciones $\{f_1, f_2, \dots, f_5\}$ y una leyenda o texto que indique el valor de ϵ .

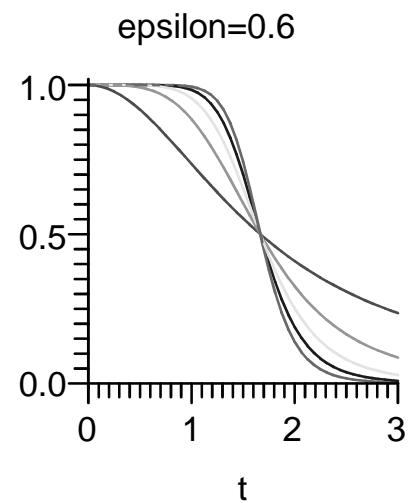
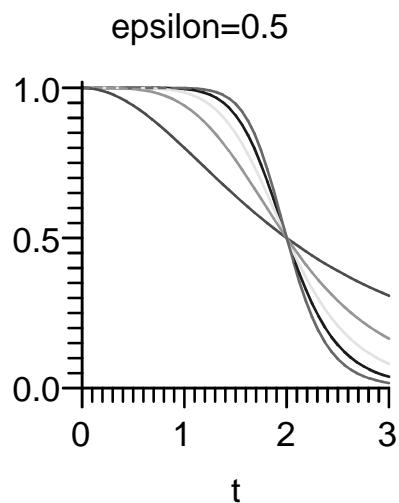
▼ Solución

Hacemos el mosaico:

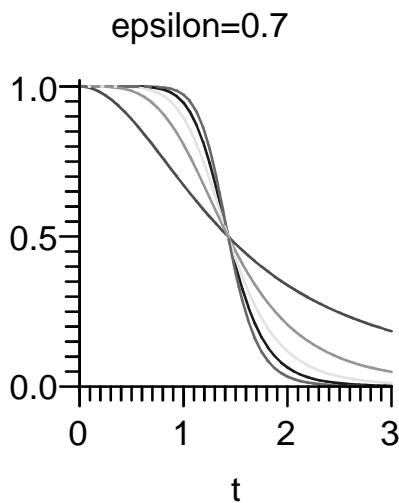
$$> \text{with}(\text{plots}) : \text{with}(\text{plottools}) :$$

$$\begin{aligned} &\left| \begin{aligned} &\text{display} \\ &\text{plot}([\text{seq}(f(t, i, 0.5), i = 1 .. 5)], t \\ &= 0 .. 3), \text{title} = "epsilon=0.5"; \end{aligned} \right| \end{aligned}$$

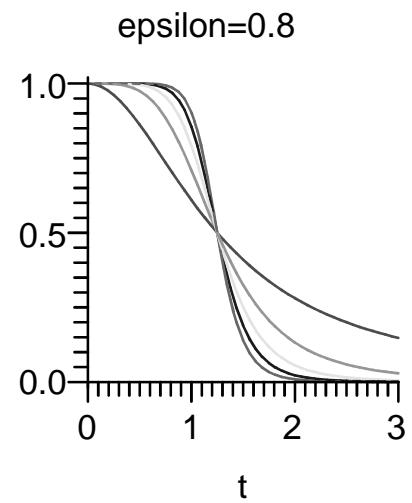
$$\begin{aligned} &\left| \begin{aligned} &\text{display} \\ &\text{plot}([\text{seq}(f(t, i, 0.6), i = 1 .. 5)], t \\ &= 0 .. 3), \text{title} = "epsilon=0.6"; \end{aligned} \right| \end{aligned}$$



```
display(  
plot( [seq(f(t, i, 0.7), i = 1 ..5 )], t  
= 0 ..3 ), title = "epsilon=0.7");
```

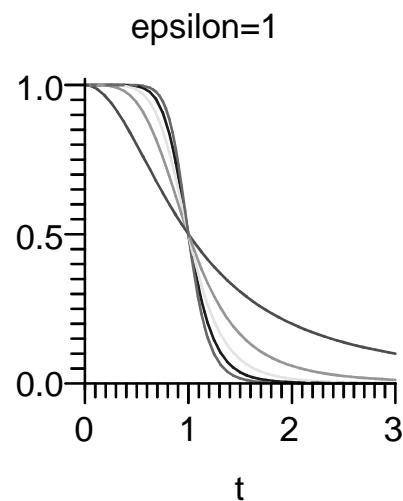
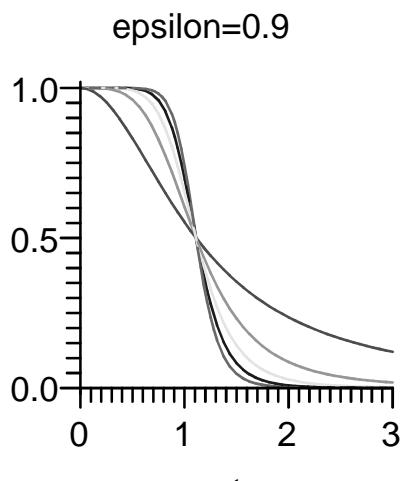


```
display(  
plot( [seq(f(t, i, 0.8), i = 1 ..5 )], t  
= 0 ..3 ), title = "epsilon=0.8");
```



```
display(  
plot( [seq(f(t, i, 0.9), i = 1 ..5 )], t  
= 0 ..3 ), title = "epsilon=0.9");
```

```
display(  
plot( [seq(f(t, i, 1 ), i = 1 ..5 )], t =  
0 ..3 ), title = "epsilon=1");
```



Apartado (c)

Comentar el efecto de n y ϵ sobre la forma de la gráfica.

Solución

El valor de n es el **orden del filtro** (Paso Bajo en este caso). Cuanto mayor es el orden de un filtro, más abrupta es su caída entre la banda de paso y la banda atenuada. El valor de ϵ está relacionado con el **ancho de banda** del filtro (en este ejercicio en dominio temporal). Cuanto menor es ϵ , más se prolonga la banda de paso, y viceversa.

Apartado (d)

Representar la señal $x(t)$ y su transformada de Fourier $X(j\omega)$.

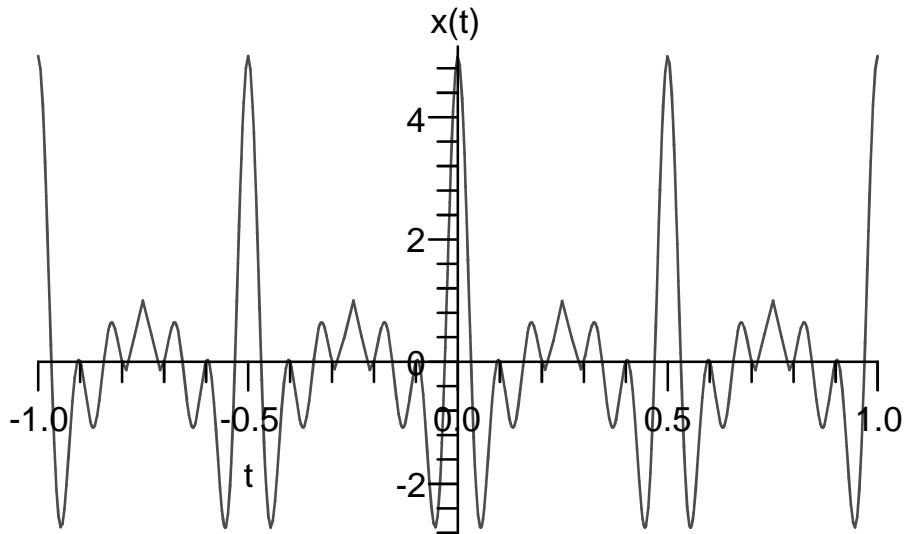
Solución

Definimos $x(t)$:

```

> x := t → ∑_{n=1}^5 cos(4 · (n + 1)π · t) :
> plot(x(t), t = -1 .. 1, title = "x(t)");

```



```
> with(inttrans):
```

Definimos la Transformada de Fourier de $x(t)$:

```
> X := omega -> fourier(x(t), t, omega):
```

```
> X(omega);
```

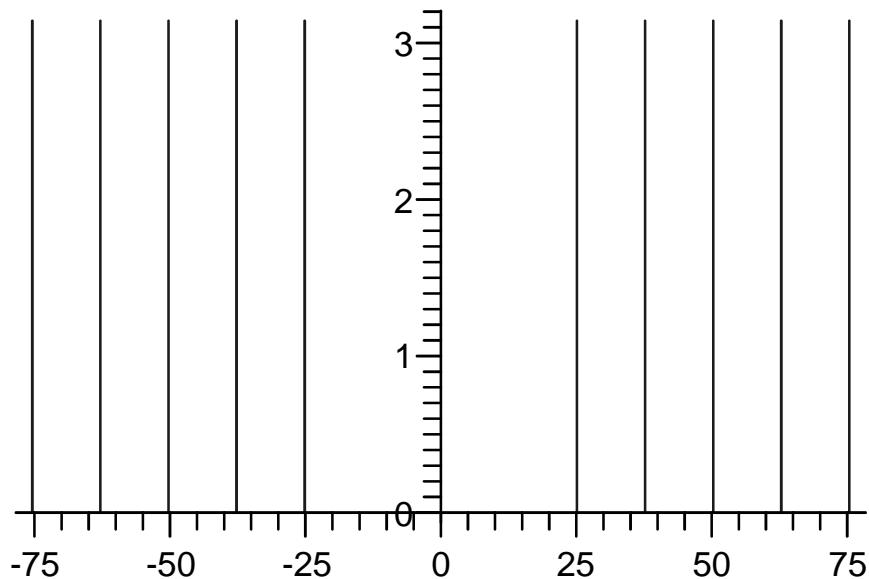
$$\begin{aligned} \pi (\text{Dirac}(\omega - 16\pi) + \text{Dirac}(\omega + 8\pi) + \text{Dirac}(\omega - 12\pi) + \text{Dirac}(\omega - 20\pi) \\ + \text{Dirac}(\omega + 20\pi) + \text{Dirac}(\omega - 8\pi) + \text{Dirac}(\omega + 12\pi) + \text{Dirac}(\omega + 24\pi) \\ + \text{Dirac}(\omega + 16\pi) + \text{Dirac}(\omega - 24\pi)) \end{aligned} \quad (5.1.1)$$

Obtenemos que $X(j\omega)$ es una suma de deltas, todas ellas de peso π . Ya que la Delta de Dirac **no es una función**, sino una *distribución* (el comando *plot* no nos vale), las representamos como líneas a su correspondiente frecuencia con peso π .

```
> plotX := {seq(line([ (4 + 4*i)*pi, 0], [ (4 + 4*i)*pi, pi]), i = 1 .. 5), seq(line([ - (4 + 4*i)*pi, 0], [ - (4 + 4*i)*pi, pi]), i = 1 .. 5)}:
```

```
> display(plotX, color = blue, title = "Transformada de Fourier x(t)");
```

Transformada de Fourier $x(t)$



Apartado (e)

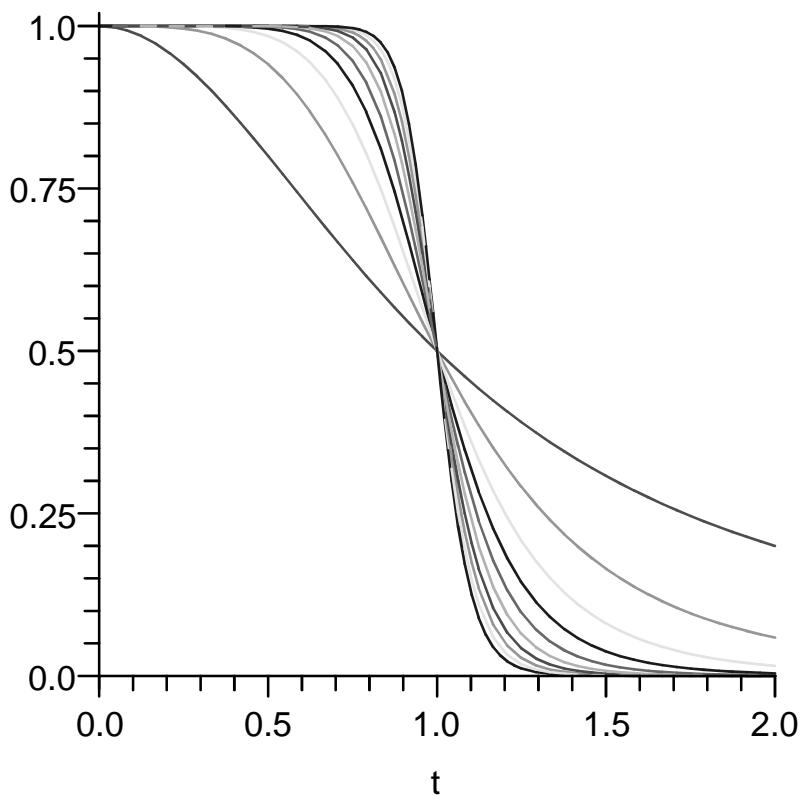
Fijar, en lo que sigue, $\epsilon = 1$ y dibujar los diez primeros elementos de la sucesión $\{f_n\}$. ¿Cuál es el punto de corte, si existe, de estas gráficas? ¿Cómo afecta el valor de ϵ al punto de intersección?

Solución

Dibujamos la sucesión:

```
> display(plot([seq(f(t, i, 1), i=1..10)], t=0..2), title="epsilon=1");
```

epsilon=1



Efectivamente, existe un punto de corte para todas las gráficas, y es el $\left[1, \frac{1}{2} \right]$. Según vimos en el apartado (a), podemos concluir que el punto de corte dada una sucesión $\{f_n\}$ es siempre de $\left[\frac{1}{\epsilon}, \frac{1}{2} \right]$. Por ello, cuanto menor es ϵ , más se ensancha la banda de paso del filtro.

Apartado (f)

Representar en un mismo gráfico el filtro paso-bajo ideal de pulsación de corte $\omega_c = \frac{50}{s} \text{ rad}$ y la aproximación correspondiente a un filtro de **Butterworth** de orden $n = 2$. ¿Cuál es el error?

Solución

Representamos el filtro Paso Bajo ideal (pulso de amplitud unidad entre $-\omega_c$ y ω_c):

```
> filtroideal := omega → piecewise(omega < 50 and omega > -50, 1):
```

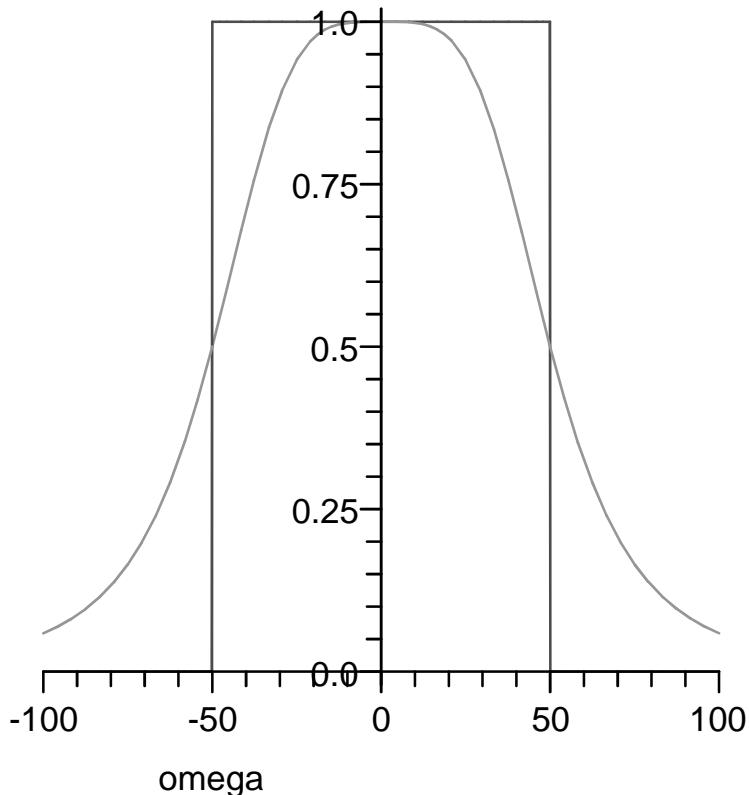
Un filtro Butterworth de orden $n = 2$ es una función del conjunto $\{f_n\}$ descrita anteriormente,

para $n = 2$, $\epsilon = 1$ y $t = \frac{\omega}{50}$. Para comparar con el filtro ideal, supondremos que tiene una amplitud unidad:

```
> butterworth := omega → f((omega/50, 2, 1):
```

```
> plot([filtroideal(omega), butterworth(omega)], omega = -100 .. 100, title = "Paso Bajo ideal y Butterworth");
```

Paso Bajo ideal y Butterworth

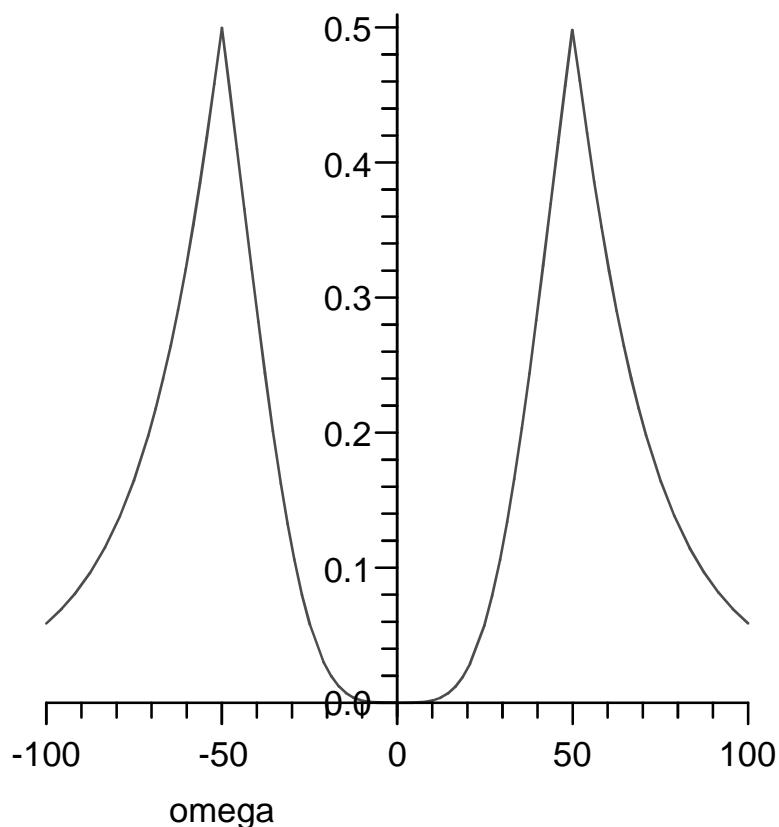


El error del filtro de Butterworth frente al filtro Paso Bajo ideal viene dado por la diferencia de valores en el eje de ordenadas.

```
> errorbutterworth := omega → abs(filtroideal(omega) - butterworth(omega)):
```

```
> plot(errorbutterworth(omega), omega = -100 .. 100, title = "Error Filtro Butterworth");
```

Error Filtro Butterworth



Como podemos ver en la gráfica del error, alcanza su valor máximo en la frecuencia de corte del filtro ideal, $\omega_c = 50$ (y es la mitad de la amplitud, es decir, $\frac{1}{2}$).

Apartado (g)

Dibujar el diagrama polos-ceros y hallar la función de transferencia $H(s)$ del filtro **Butterworth** para $n = 2$.

Solución

Los filtros de **Butterworth**, por definición, no tienen *ceros*.

Para determinar la localización de los polos en el plano complejo, acudimos a la expresión teórica de los polos en un filtro **Butterworth**:

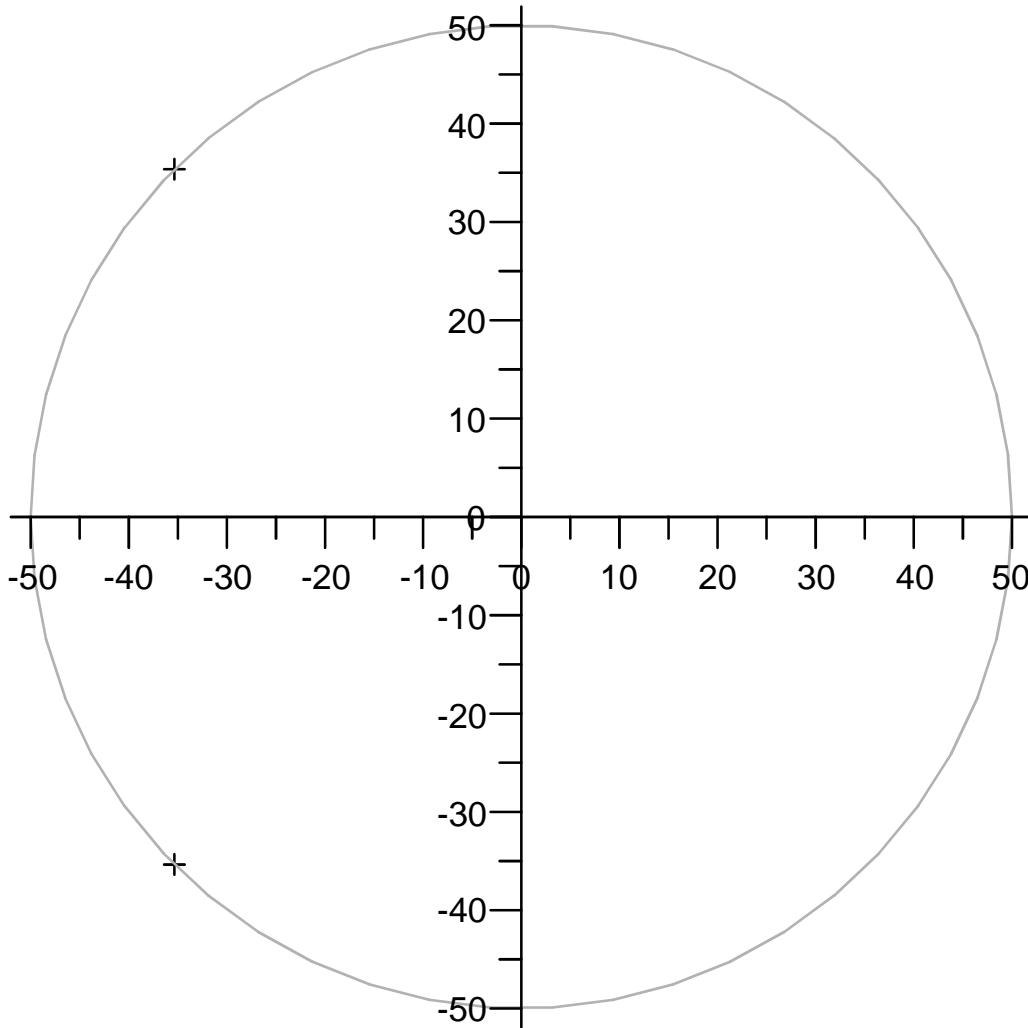
$$\frac{-s_k^2}{\omega_c^2} = (-1)^n = e^{-\frac{j(2k-1)\pi}{n}} \text{ para } k = 1, 2 \dots n.$$

Por tanto:

$$\begin{aligned} & s := (\omega_c, n, k) \rightarrow \omega_c \cdot e^{\frac{j \cdot (2k+n-1) \cdot \pi}{2 \cdot n}} : \\ & \text{polos} := [\text{seq}(s(50, 2, i), i=1..2)]; \\ & \text{polos} := [-25\sqrt{2} + 25I\sqrt{2}, -25\sqrt{2} - 25I\sqrt{2}] \end{aligned} \quad (8.1.1)$$

Obtenemos que todos los polos se encuentran en la circunferencia de radio 50 y con fase $\left\{ \frac{3\pi}{4}, \frac{5\pi}{4} \right\}$.

```
> display(pointplot( {seq( [abs(polos[i]), argument(polos[i])], i=1..2) }, coords=polar, symbol=cross, color=red), circle([0,0], 50), color=aquamarine);
```



Ahora procedemos a escribir la función de transferencia $H(s)$. Sabemos que, conocidos los polos s_k :

$$H(s) = \frac{1}{\prod_{k=1}^n (s - s_k)}$$

Por tanto:

$$\begin{aligned} &> H := s \rightarrow \frac{1}{\prod_{k=1}^{\text{nops}(polos)} (s - \text{polos}[k])}; \\ &\qquad H := s \rightarrow \frac{1}{\prod_{k=1}^{\text{nops}(polos)} (s - \text{polos}_k)} \end{aligned} \tag{8.1.2}$$

$$> H(s); \tag{8.1.3}$$

$$\frac{1}{(s + 25\sqrt{2} - 25\text{i}\sqrt{2})(s + 25\sqrt{2} + 25\text{i}\sqrt{2})} \quad (8.1.3)$$

Apartado (h)

Hallar la respuesta impulsional $h(t)$ de los sistemas correspondientes a los filtros paso-bajo ideal y **Butterworth**.

Solución

Calculamos la respuesta al impulso $h(t)$ para el filtro paso-bajo ideal:

$$> hfiltroideal := \text{invfourier}(\text{filtroideal}(\omega), \omega, t); \\ hfiltroideal := \frac{\sin(50t)}{t\pi} \quad (9.1.1)$$

Calculamos la respuesta al impulso $h(t)$ del filtro **Butterworth**:

$$> hbutterworth := \text{invlaplace}(H(s), s, t); \\ hbutterworth := \frac{1}{50}\sqrt{2} e^{(-25\sqrt{2}t)} \sin(25\sqrt{2}t) \quad (9.1.2)$$

Apartado (i)

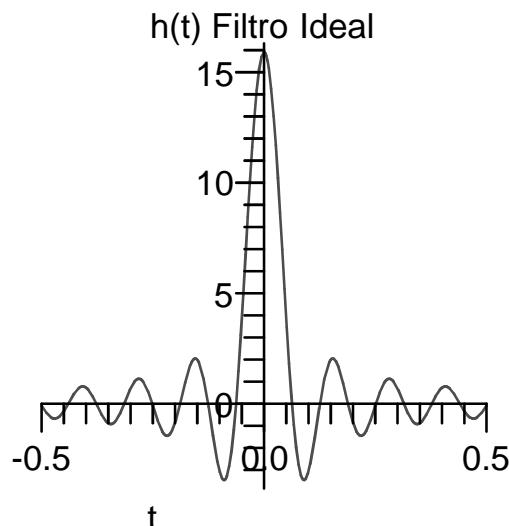
Dibujar en un mosaico fila las respuestas impulsionales

Solución

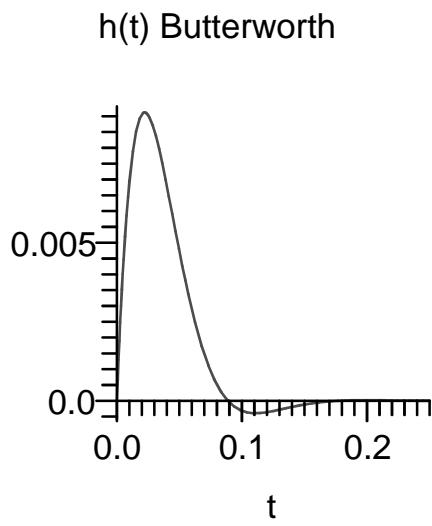
Representamos el mosaico:

Vemos que la respuesta al impulso del filtro ideal es una función **sinc**, definida en todo el dominio temporal (puesto que en el dominio de la frecuencia estaba acotada), y a la inversa con el filtro **Butterworth**, que estaba definido para toda pulsación, y ahora se encuentra acotado en el dominio temporal.

```
> plot(hfiltroideal, t = -0.5 .. 0.5, title = "h(t) Filtro Ideal");
```



```
> plot(hbutterworth(t), t = 0 .. 0.25, title = "h(t) Butterworth");
```



Apartado (j)

Obtener la salida, $y(t)$, de los filtros anteriores tanto en el dominio temporal como en el frecuencial.

▼ Solución

Primero calculamos la salida para el filtro Paso Bajo ideal:

Al ser un filtrado ideal, se eliminan las componentes de $x(t)$ (deltas) para pulsaciones mayores a $\frac{50 \text{ rad}}{s}$:

```
> Yfiltroideal := X(ω) · (Heaviside(ω + 50) - Heaviside(ω - 50)) :
```

```
> yfiltroideal := invfourier(Yfiltroideal, ω, t) :
```

Y la salida de $x(t)$ para un filtro Butterworth:

```
> Ybutterworth := X(ω) · butterworth(ω) :
```

```
> ybutterworth := invfourier(Ybutterworth, ω, t) :
```

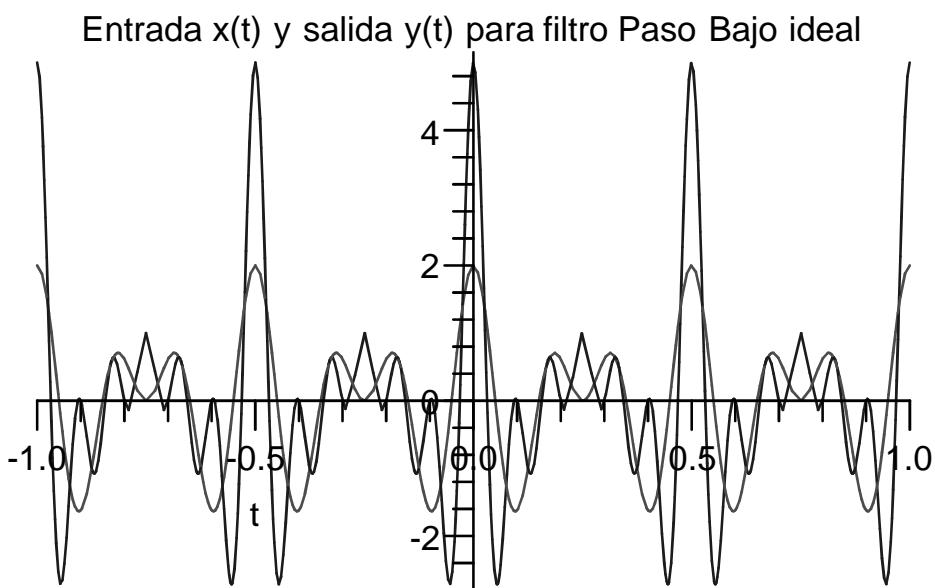
▼ Apartado (k)

Representar en un mismo gráfico la entrada $x(t)$ y la salida $y(t)$ para los casos considerados y comentar los resultados:

▼ Solución

Para el filtro Paso Bajo **ideal**:

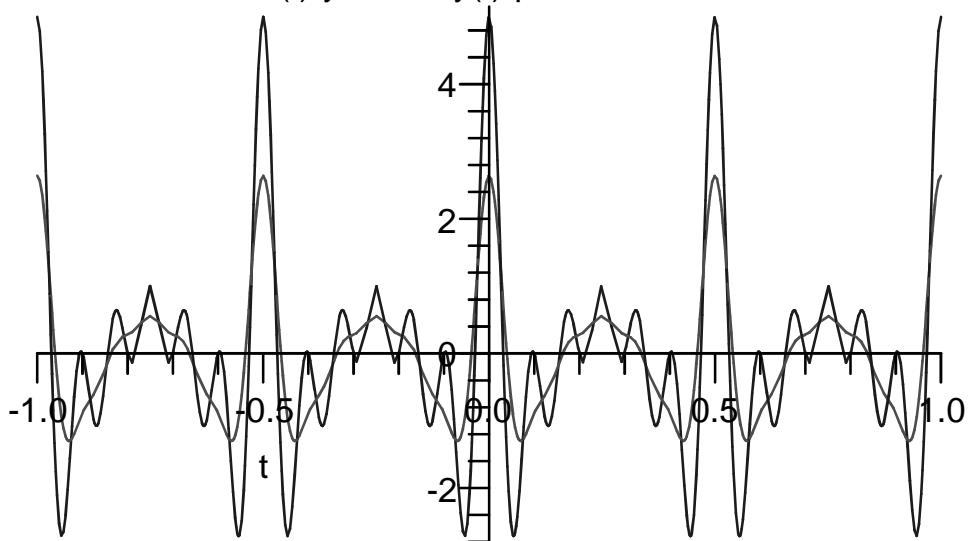
```
> display( [plot(x(t), t = -1 .. 1, color = blue), plot(yfiltroideal, t = -1 .. 1, color = red)],  
           title = "Entrada x(t) y salida y(t) para filtro Paso Bajo ideal");
```



Para el filtro Paso Bajo **Butterworth**:

```
> display( [plot(x(t), t = -1 .. 1, color = blue), plot(ybutterworth, t = -1 .. 1, color = red)],  
           title = "Entrada x(t) y salida y(t) para filtro Butterworth");
```

Entrada $x(t)$ y salida $y(t)$ para filtro Butterworth



Podemos ver la diferencia entre las dos señales $y(t)$ resaltadas en rojo: mientras que en el filtrado ideal se mantiene el aspecto sinusoidal de la señal, en el filtrado con Butterworth se produce una distorsión.

▼ Apartado (I)

Repetir los cálculos anteriores para el filtro Butterworth de orden $n = 4$.

▼ Solución

Repetimos los apartados anteriores para el filtro **Butterworth** con $n = 4$.

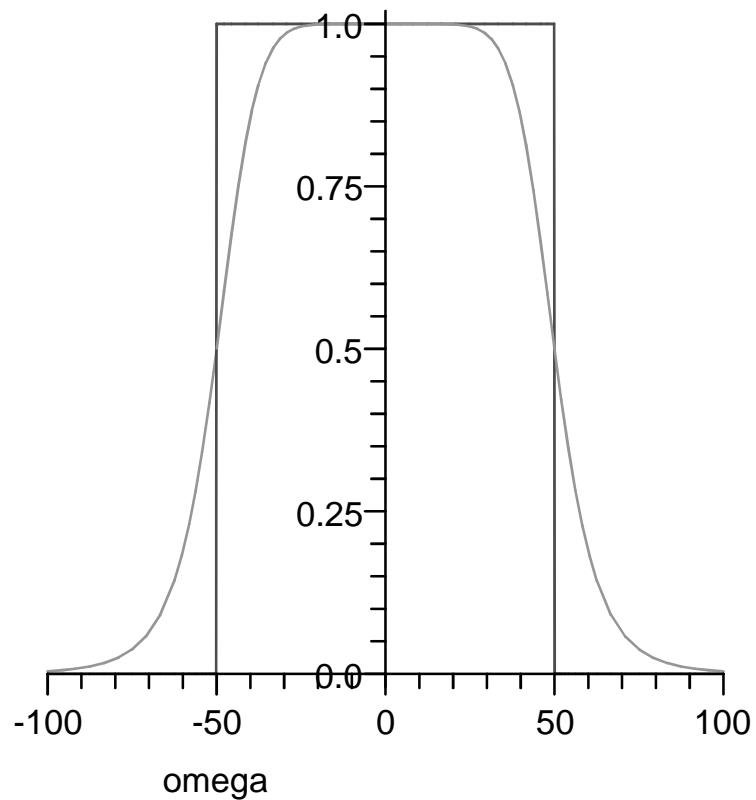
▼ Representación en un gráfico junto al paso-bajo ideal:

Un filtro Butterworth de orden $n = 4$ es una función del conjunto $\{f_n\}$ descrita

anteriormente, para $n = 4$, $\epsilon = 1$ y $t = \frac{\omega}{50}$. Para comparar con el filtro ideal, supondremos que tiene una amplitud unidad:

```
> butterworth4 := omega -> f((omega/50), 4, 1):
> plot([filtroideal(omega), butterworth4(omega)], omega = -100..100, title =
    "Paso Bajo ideal y Butterworth n=4");
```

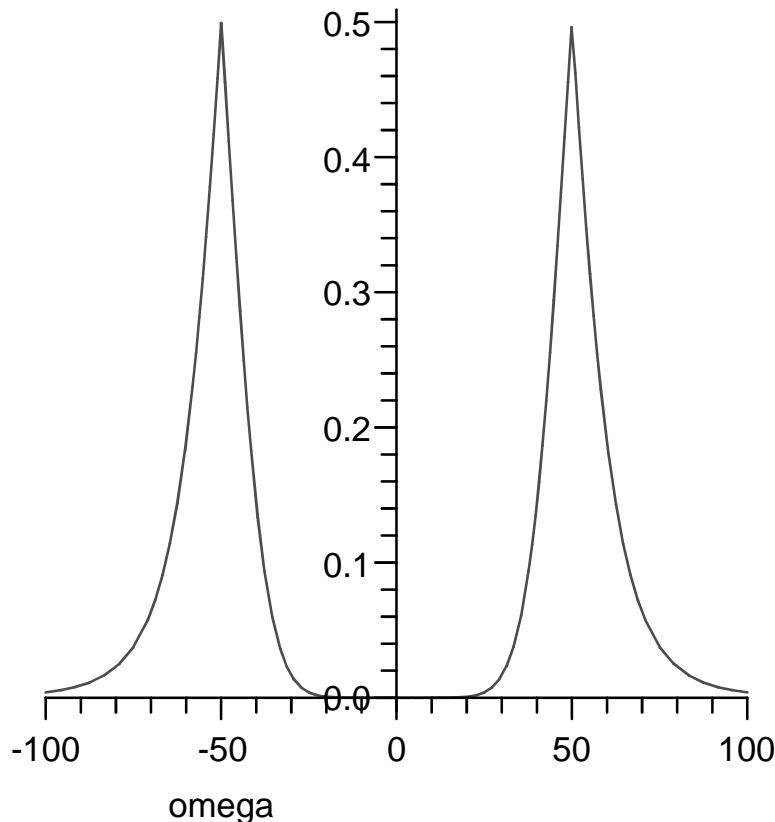
Paso Bajo ideal y Butterworth n=4



El error del filtro de Butterworth frente al filtro Paso Bajo ideal viene dado por la diferencia de valores en el eje de ordenadas.

```
[> errorbutterworth4 :=  $\omega \rightarrow \text{abs}(\text{filtroideal}(\omega) - \text{butterworth4}(\omega))$ ;  
> plot(errorbutterworth4( $\omega$ ),  $\omega = -100 .. 100$ , title = "Error Filtro Butterworth n=4");
```

Error Filtro Butterworth n=4



Como podemos ver en la gráfica del error, alcanza su valor máximo en la frecuencia de corte del filtro ideal, $\omega_c = 50$ (y es la mitad de la amplitud, es decir, $\frac{1}{2}$). Sin embargo, la diferencia principal de esta función de error para $n = 4$ es la reducción del ancho en los picos de la función.

Dibujar el diagrama de polos y ceros:

Los filtros de **Butterworth**, por definición, no tienen *ceros*.

Para determinar la localización de los polos en el plano complejo, acudimos a la expresión teórica de los polos en un filtro **Butterworth**:

$$\frac{-s_k^2}{\omega_c^2} = (-1)^{\frac{1}{n}} = e^{-\frac{j(2k-1)\pi}{n}} \quad \text{para } k = 1, 2 \dots n.$$

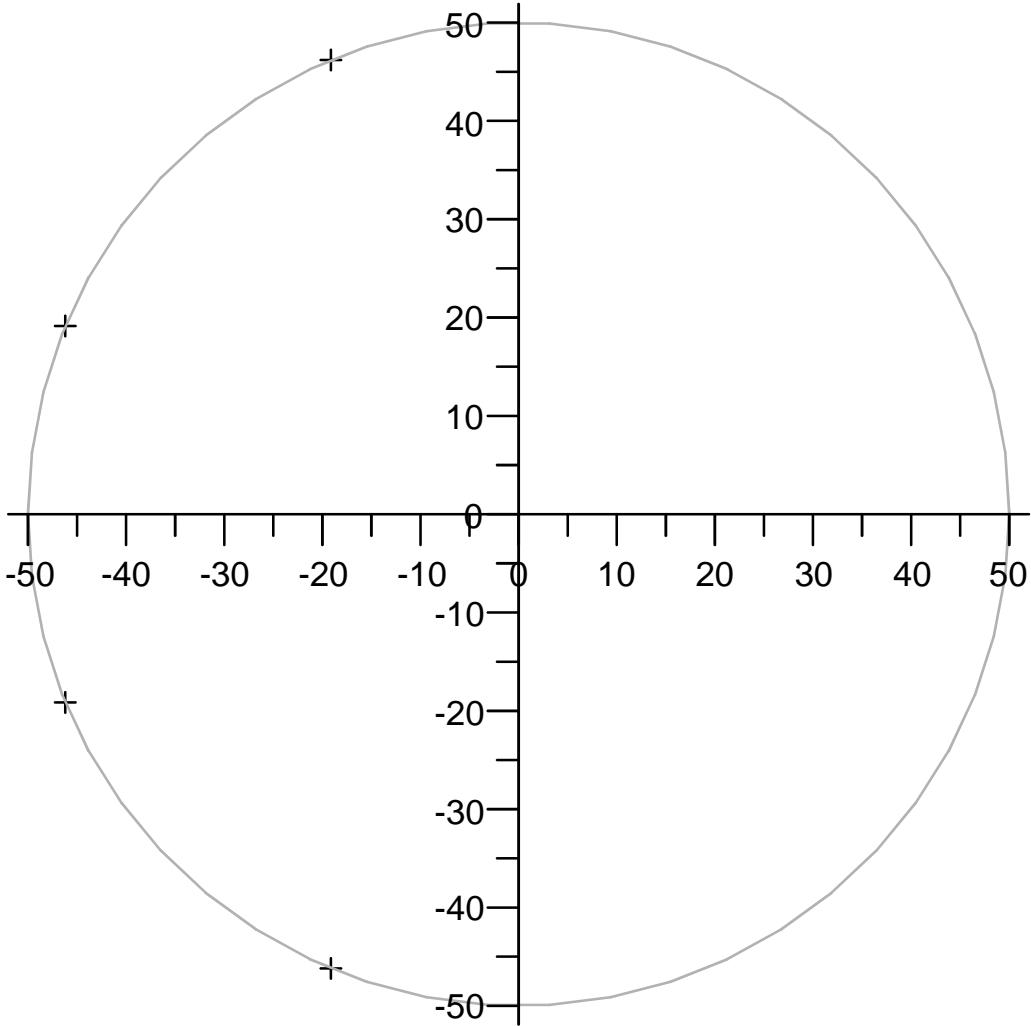
Por tanto:

$$\begin{aligned} > \text{polos4} := [\text{seq}(s(50, 4, i), i = 1 .. 4)]; \\ & \text{polos4} := \left[50 e^{\left(\frac{5}{8} i\pi\right)}, 50 e^{\left(\frac{7}{8} i\pi\right)}, 50 e^{\left(-\frac{7}{8} i\pi\right)}, 50 e^{\left(-\frac{5}{8} i\pi\right)} \right] \end{aligned} \quad (13.1.2.1)$$

Obtenemos que todos los polos se encuentran en la circunferencia de radio 50 y con fase $\left\{ \frac{5\pi}{8}, \frac{7\pi}{8}, -\frac{7\pi}{8}, -\frac{5\pi}{8} \right\}$.

$$\begin{aligned} > \text{display}(\text{pointplot}(\{\text{seq}([\text{abs}(\text{polos4}[i]), \text{argument}(\text{polos4}[i])], i = 1 .. 4)\}, \\ & \quad \text{coords} = \text{polar}, \text{symbol} = \text{cross}, \text{color} = \text{red}), \text{circle}([0, 0], 50), \text{color} = \end{aligned}$$

aquamarine);



Ahora procedemos a escribir la función de transferencia $H(s)$. Sabemos que, conocidos los polos s_k :

$$H(s) = \frac{1}{\prod_{k=1}^n (s - s_k)}$$

Por tanto:

$$\begin{aligned} > H2 := s \rightarrow \frac{1}{\prod_{k=1}^{\text{nops}(polos4)} (s - \text{polos4}[k])}; \\ & H2 := s \rightarrow \frac{1}{\prod_{k=1}^{\text{nops}(polos4)} (s - \text{polos4}_k)} \end{aligned} \quad (13.1.2.2)$$

$$\begin{aligned} > H2(s); \\ & \frac{1}{\left(s - 50 e^{\left(\frac{5}{8} i\pi\right)}\right) \left(s - 50 e^{\left(\frac{7}{8} i\pi\right)}\right) \left(s - 50 e^{\left(-\frac{7}{8} i\pi\right)}\right) \left(s - 50 e^{\left(-\frac{5}{8} i\pi\right)}\right)} \end{aligned} \quad (13.1.2.3)$$

Hallar la respuesta al impulso $h(t)$

Calculamos la respuesta al impulso $h(t)$ del filtro **Butterworth** con $n = 4$.

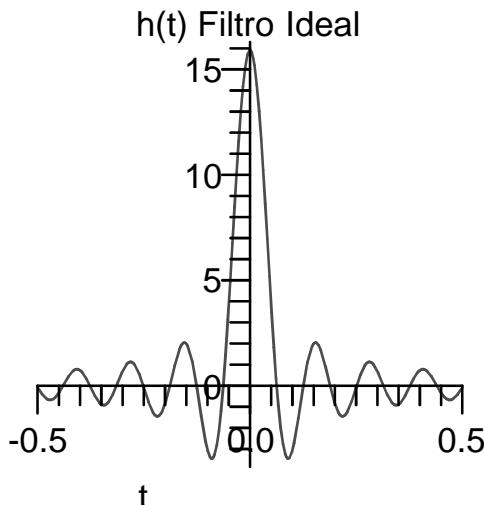
[> $hbutterworth4 := \text{invlaplace}(H2(s), s, t)$:

Dibujar un mosaico con las dos respuestas impulsionales

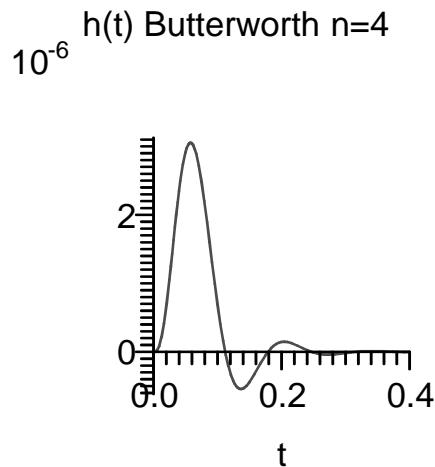
Representamos el mosaico:

Vemos que la respuesta al impulso del filtro ideal es una función **sinc**, definida en todo el dominio temporal (puesto que en el dominio de la frecuencia estaba acotada), y a la inversa con el filtro **Butterworth**, que estaba definido para toda pulsación, y ahora se encuentra acotado en el dominio temporal. Como el orden del filtro Butterworth es mayor, su Transformada de Fourier se asemeja más al filtro paso-bajo ideal, y por tanto, su respuesta temporal se va pareciendo más a una **función sinc** (se van formando más armónicos).

```
> plot(hfiltroideal, t = -0.5 ..0.5, title  
= "h(t) Filtro Ideal");
```



```
> plot(hbutterworth4(t), t = 0 ..0.4,  
title = "h(t) Butterworth n=4");
```



Obtener la salida $y(t)$

Calculamos la salida $y(t)$ para un filtro Butterworth con $n = 4$:

[> $Ybutterworth4 := X(\omega) \cdot \text{butterworth4}(\omega)$:

[> $ybutterworth4 := \text{invfourier}(Ybutterworth4, \omega, t)$:

Representar la salida $y(t)$ para el paso-bajo ideal y para Butterworth $n=4$

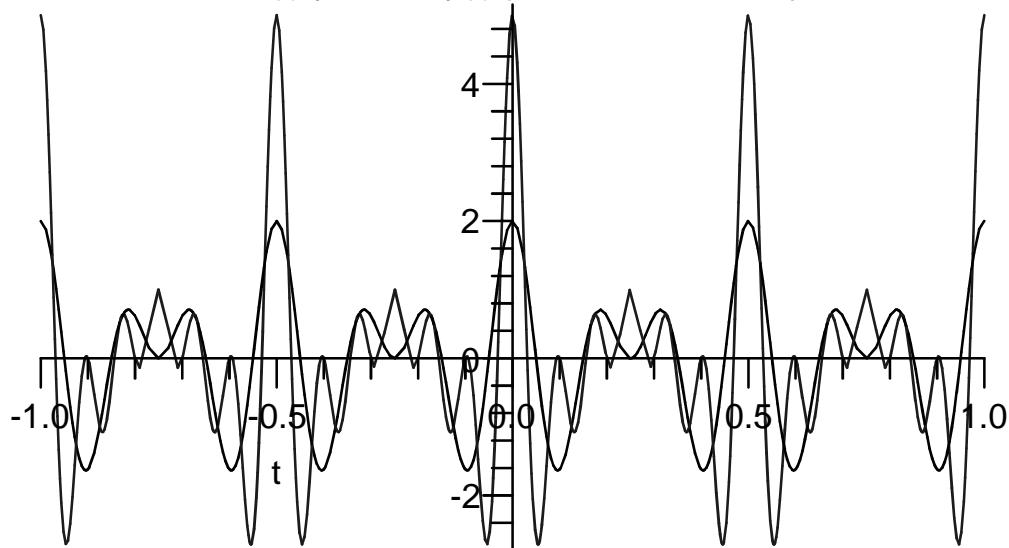
Para el filtro Paso Bajo **ideal**:

[> $plotx := \text{plot}(x(t), t = -1 ..1, \text{color} = \text{blue})$:

[> $plotideal := \text{plot}(yfiltroideal, t = -1 ..1, \text{color} = \text{black})$:

[> $\text{display}([\text{plotx}, \text{plotideal}], \text{title} =$
"Entrada x(t) y salida y(t) para filtro Paso Bajo ideal");

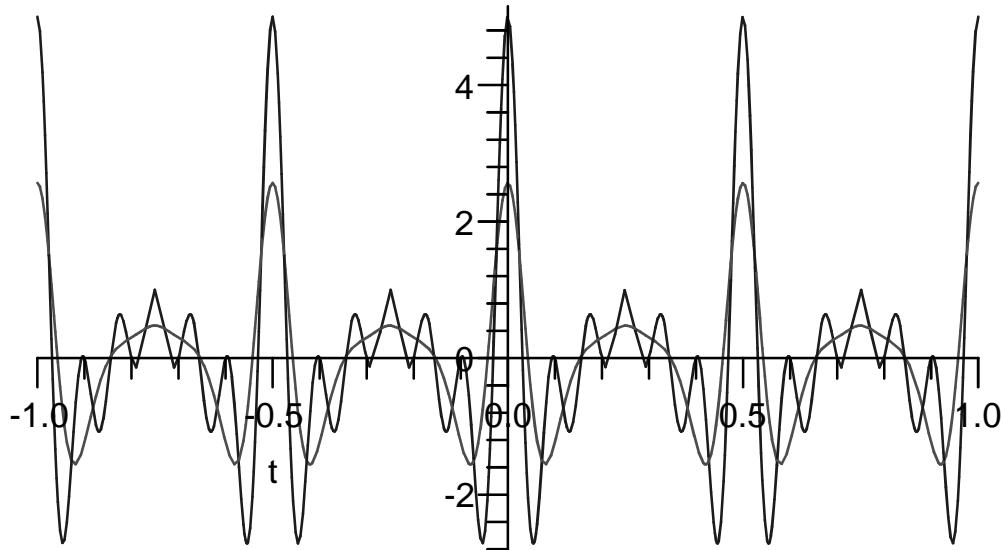
Entrada $x(t)$ y salida $y(t)$ para filtro Paso Bajo ideal



Para el filtro Paso Bajo **Butterworth con $n=4$** :

```
> plotbutterworth4 := plot(ybutterworth4, t = -1 .. 1, color = red) :  
> display([plotx, plotbutterworth4], title =  
"Entrada x(t) y salida y(t) para filtro Butterworth n=4");
```

Entrada $x(t)$ y salida $y(t)$ para filtro Butterworth $n=4$



Podemos ver la diferencia entre las dos señales $y(t)$ resaltadas en rojo: mientras que en el filtrado ideal se mantiene el aspecto sinusoidal de la señal, en el filtrado con Butterworth se produce una distorsión. Sin embargo, al haber aumentado el orden del filtro

Butterworth, la distorsión que se produce respecto al paso-bajo ideal es algo más suave, aunque todavía no se consigue la forma de la señal filtrada idealmente.

Apartado (m)

Determinar el valor mínimo de n para que las salidas ideal y real, $y(t)$, coincidan visualmente.

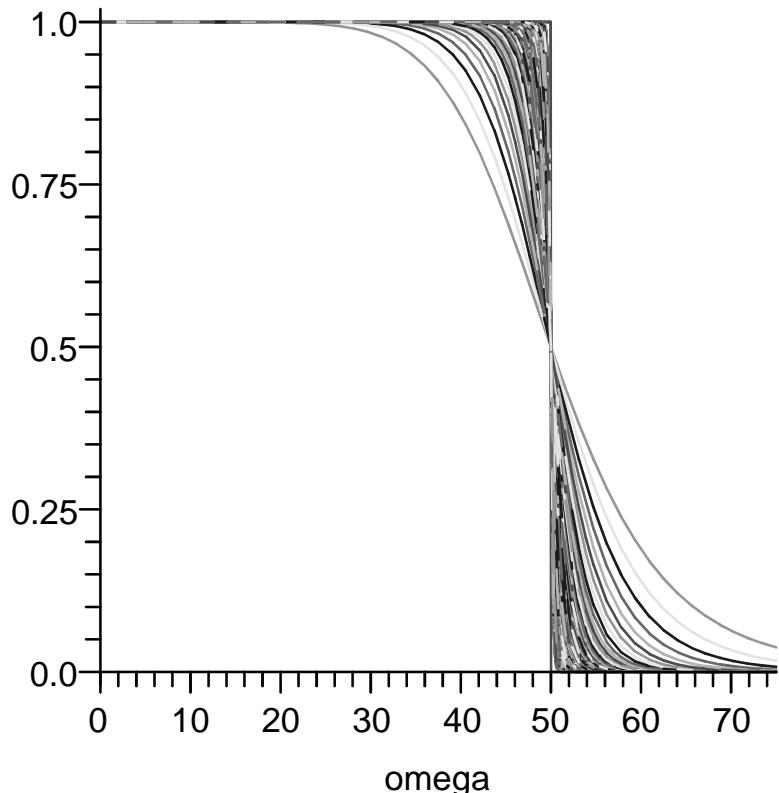
Solución

Está claro que para que las salidas $y(t)$ real e ideal tengan un aspecto similar, tenemos que incrementar el orden del filtro lo máximo posible: el caso en que el filtro **Butterworth** sea

idéntico al ideal será para $n \rightarrow \infty$; haremos una aproximación haciendo n muy grande: Representamos en el dominio de la frecuencia el aspecto de filtros **Butterworth** con orden de $n = 4$ a $n = 200$, y podemos comprobar que la curva se approxima asintóticamente al filtro paso -bajo ideal.

```
> plot([filtroideal(omega), seq(f((omega/50)^i, i, 1), i=4..200)],  
      omega=0..75, title="Paso Bajo ideal y Butterworth de n=4 a n=200");
```

Paso Bajo ideal y Butterworth de n=4 a n=200



Ahora bien, pensemos en la Transformada de Fourier de la señal $x(t)$, es decir, $X(\omega)$. Este espectro posee ciertas deltas en $\{\omega = 8\pi, \omega = 12\pi, \omega = 16\pi, \omega = 20\pi \text{ y } \omega = 24\pi\}$ en topología unilateral. Hallando sus valores exactos en ω :

```
> {seq((4 + 4 * i) * pi, i = 1 .. 5)};  
{8 pi, 12 pi, 16 pi, 20 pi, 24 pi} (14.1.1)
```

```
> deltas := evalf(%);  
deltas := {50.26548246, 37.69911185, 75.39822370, 25.13274123, 62.83185308} (14.1.2)
```

Con el filtro paso-bajo ideal eliminamos las componentes frecuenciales tal que $\omega > \omega_c$. Como $\omega_c = 50$, tenemos que sólo nos quedamos con las deltas $\omega < \omega_c$:

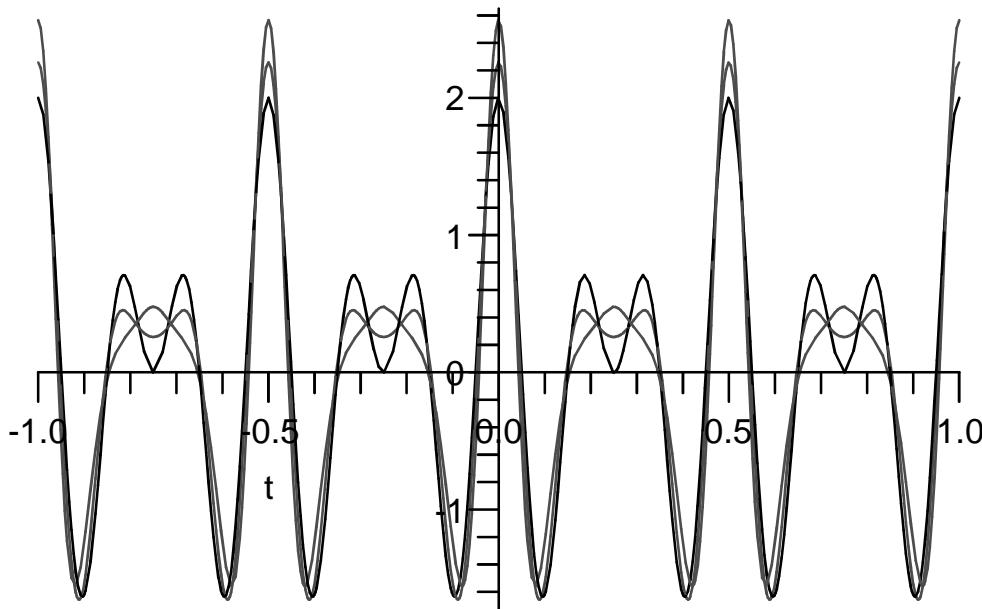
```
> select(x -> evalb(x < 50), deltas);  
{37.69911185, 25.13274123} (14.1.3)
```

En resumen, siempre que consigamos obtener una respuesta en frecuencia similar a esas dos

deltas con un **filtro Butterworth de orden n**, estaremos aproximándonos al filtrado ideal. Podemos visualizar en el dominio del tiempo el aspecto que sufre la salida $y(t)$ conforme aumentamos el orden del filtro Butterworth.

En concreto, visualizaremos la salida $y(t)$ con filtrado ideal, y la salida $y(t)$ para ordenes de Butterworth muy altos ($n = 200, n = 300 \dots$).

```
> display([plotideal, seq(
  plot(invfourier((X(omega)·f((omega/50), i, 1)), omega, t), t = -1..1), i = [4, 100])]);
```



Podemos ver que para un $n = 300$ la salida $y(t)$ tiene una apariencia idéntica a la $y(t)$ filtrada idealmente, luego podemos considerar $n = 300$.

Apartado (n)

Calcular, tanto de manera exacta como aproximada, para $n = 10$ y $\epsilon = 0.8$ las raíces de $1 + (\epsilon t)^{2n}$, realizar el diagrama polos-ceros correspondiente y calcular la función de transferencia por ambos métodos.

Solución

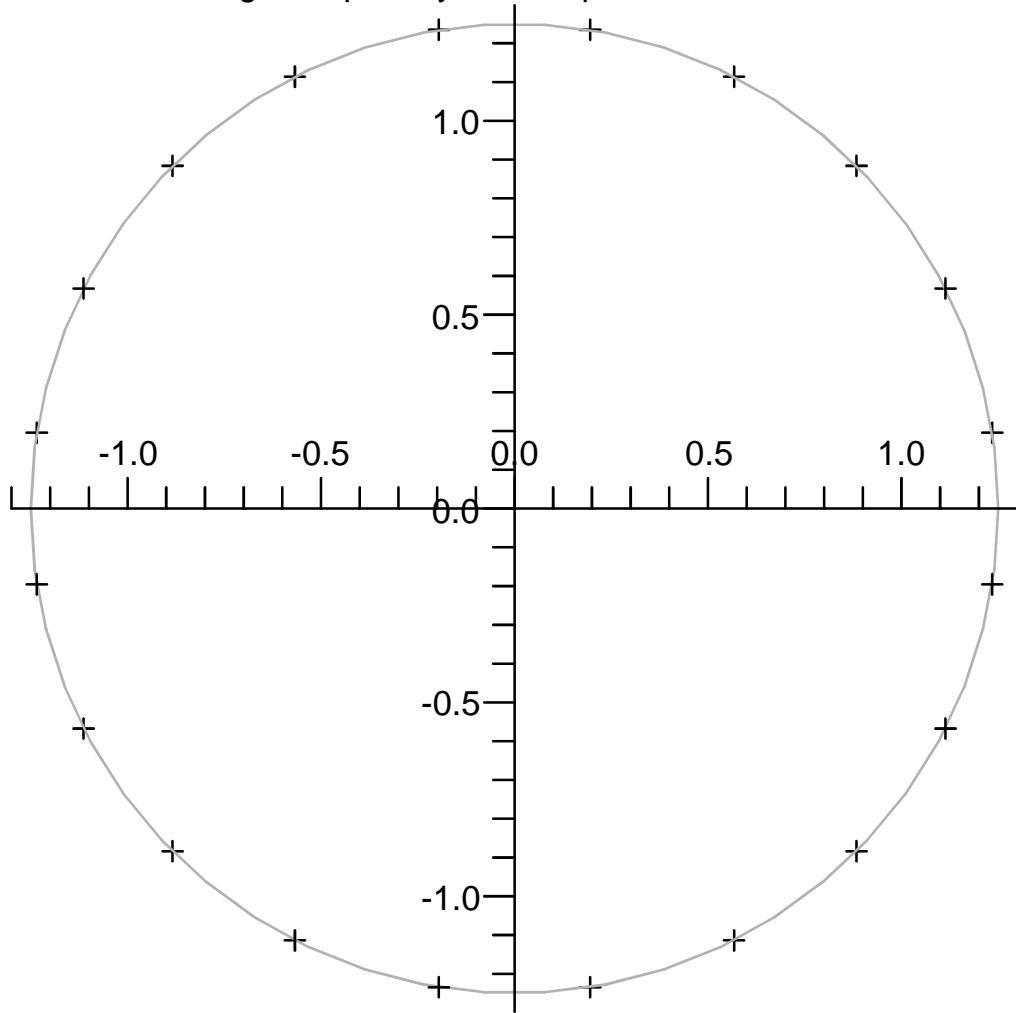
Distinguimos los dos métodos

Cálculo exacto

Ayudándonos de las herramientas que nos ofrece Maple:

```
> raices := [solve(1 + (0.8 t)^2 · 10 = 0)];
> circulo := circle([0, 0], abs(raices[1]), color = aquamarine):
> display({circulo, seq(
  pointplot([abs(raices[i]), argument(raices[i])])
  , coords = polar, symbol = cross), i = 1 .. nops(raices))})
, title = "Diagrama polos y ceros epsilon=0.8, n=10");
```

Diagrama polos y ceros epsilon=0.8, n=10



Y su función de transferencia $H(s)$, estará formada por el conjunto de polos situados a la izquierda del plano complejo, es decir, aquellos cuya parte real sea negativa.

$\begin{array}{|l} > \text{raicesH} := \text{select}(x \rightarrow \text{evalb}(\Re(x) < 0), \text{raices}) : \\ \end{array}$

$$\begin{array}{|l} > H3 := s \rightarrow \frac{1}{\prod_{k=1}^{\text{nops(raicesH)}} (s - \text{raicesH}[k])} : \\ \end{array}$$

$\begin{array}{|l} > H3(s); \\ \end{array}$

$$\begin{aligned} & 1 / ((s + 0.1955430813 - 1.234610426 I) (s \\ & + 0.5674881247 - 1.113758155 I) (s \\ & + 0.8838834765 - 0.8838834765 I) (s \\ & + 1.113758155 - 0.5674881247 I) (s \\ & + 1.234610426 - 0.1955430813 I) (s + 1.234610426 \\ & + 0.1955430813 I) (s + 1.113758155 + 0.5674881247 I) (s \\ & + 0.8838834765 + 0.8838834765 I) (s + 0.5674881247 \\ & + 1.113758155 I) (s + 0.1955430813 + 1.234610426 I)) \end{aligned} \quad (15.1.1.1)$$

Método de Kasiski

Ruben Darias Bello - Luciano Rubio Romero
Sistemas de Cálculo Simbólico
Grupo 0416, Curso 2007/08

▼ Enunciado

Sabiendo que el siguiente criptograma –sin signos de puntuación ni saltos de línea–

FCSPRRXGUSBWDCQDASVIYWDVDJANWGVGBXVEGGGMJGHVVGF
OIVQGYMUGGDLGUHHALCTHZJQGHVYTCVIFFCFIVCJHHWNDUMUKC
GMUQASZSGZXTLKARKGODUIVQFWHCURXGTSOTSOSQWKFSWZWUQ
LMFVCVUANTUIFECVLMTOQBWGHBAGASWDCFVWFPSDCMPWFWIWS
DXSTSFQSCHLBMNCGMJDUMKGBWIFVSGMDQGVCUGGLDGUDUWHKS
WIJKCVBJCPDRSCZRAAPEXQDKBRAKGBHOSDOGMKRWDLSFOPMFVS
DZWPCYIJNCVIJSQLSOHWLQGDUWPCVYMGQRVKKBWQWTOQMFHC
UUAFOETWUGXJAFOVLWCZTCANSUTGUWQYMKZLVGUEXMKGCOQSPZ
DXJQLLUSGLSZGRWDKAQBGHALCPDVGGHXWTOWKCQDJSDOQXGTO
FMHVOUTSUIEQVCGRJJGHRLGEIDVVQZDZSIFHOSDOFWFCWUMUQB
FQDKOGWJSIHTSUIEQVCGHZACTLKLKQLIVWFDVLGZRAUKBFWHTW
PMJQGPMKGGHVUWOQBGCRRAPEXTAPCVYMGGHXMUWHZGPRXZGUT
XMJQBVKVWWCAFcvxgtddvacuxivqgdymksqmkusgqgczrrsow
HVLQUUILKGBYMGTLZECFRVLQRRTGSIHAWSIAGGBHAGJIEWMPP
HVVHWFQGFCETWGZDTIWWOMJUIEQAQZDQFFSPVABOFQGPFHAWTJ
DLSCZLVIWZLTAPCSWJUIDZJGBGIEKSQBGECUZWUDRVVKODASEQ
DZVNOMIMTWDHGNO

corresponde a un cifrado **Vigènere** clásico, se pide:

▼ Apartado (a)

Calcular el número de caracteres del criptograma y asignarlo a la variable `númerocaracteres`. ¿Cómo influye este valor en el éxito del proceso de descifrado?

▼ Solución

Cargamos los paquetes de Maple necesarios para este ejercicio:

```
> with(StringTools) : with(combinat) : with(Statistics) :  
> criptograma := "  
FCSPRRXGUSBWDCQDASVIYWDVDJANWGVGBXVEGGGMJGHVVGF  
OIVQGYMUGGDLGUHHALCTHZJQGHVYTCVIFFCFIVCJHHWNDUMUKC  
GMUQASZSGZXTLKARKGODUIVQFWHCURXGTSOTSOSQWKFSWZWUQ  
LMFVCVUANTUIFECVLMTOQBWGHBAGASWDCFVWFPSDCMPWFWIWS  
DXSTSFQSCHLBMNCGMJDUMKGBWIFVSGMDQGVCUGGLDGUDUWHKS  
WIJKCVBJCPDRSCZRAAPEXQDKBRAKGBHOSDOGMKRWDLSFOPMFVS  
DZWPCYIJNCVIJSQLSOHWLQGDUWPCVYMGQRVKKBWQWTOQMFHC  
UUAFOETWUGXJAFOVLWCZTCANSUTGUWQYMKZLVGUEXMKGCOQSPZ  
DXJQLLUSGLSZGRWDKAQBGHALCPDVGGHXWTOWKCQDJSDOQXGTO  
FMHVOUTSUIEQVCGRJJGHRLGEIDVVQZDZSIFHOSDOFWFCWUMUQB  
FQDKOGWJSIHTSUIEQVCGHZACTLKLKQLIVWFDVLGZRAUKBFWHTW  
PMJQGPMKGGHVUWOQBGCRRAPEXTAPCVYMGGHXMUWHZGPRXZGUT  
XMJQBVKVWWCAFcvxgtddvacuxivqgdymksqmkusgqgczrrsow  
HVLQUUILKGBYMGTLZECFRVLQRRTGSIHAWSIAGGBHAGJIEWMPP  
HVVHWFQGFCETWGZDTIWWOMJUIEQAQZDQFFSPVABOFQGPFHAWTJ  
DLSCZLVIWZLTAPCSWJUIDZJGBGIEKSQBGECUZWUDRVVKODASEQ  
DZVNOMIMTWDHGNO"
```

WHTWPMJQGPMKGGHVUWOQBGCZRAAPEXTAPCVYMGGHXMUWHZG\\
 PRXZGUTXMJQBVKVWWCAFVXGTDDVACUXIVQGDYMKUSGQ\\
 GCZRRSOWHVLQUUILKGBYMGTLZECFRVLQRRTGSIHAWSLAGGBHAG\\
 JIEWMPPHVWHWFQGFCETWGZDTIWWOMJUIEQSAZDQFFSPVABOFQG\\
 PFHAWTJDLSCLVIWZLTAPCSWJUIDZJGBGIEKSQBGECKUZWUDRVVKO\\
 DASEQDZNOMIMTWDHGNO":

Ahora calculamos el número de caracteres del criptograma:

$$\begin{aligned}
 > \text{númerocaracteres} = \text{length}(\text{criptograma}); \\
 &\quad \text{númerocaracteres} = 815
 \end{aligned} \tag{2.1.1}$$

El criptograma tiene **815 caracteres**.

El método de Kasiski permite determinar la longitud de una clave de un cifrado Vigenère, localizando las palabras repetidas en un texto cifrado. Kasiski se percató de que la distancia entre las palabras repetidas era múltiplo de la longitud de la clave utilizada para cifrar, por tanto, calculando el máximo común divisor de esas distancias, obtendremos la longitud original o un múltiplo primo de ésta.

Tras descubrir el tamaño de la clave, se aplica un método César para descifrar el criptograma. Es por ello que cuanto **más número de caracteres** tenga nuestro criptograma (el nuestro en concreto, 815), más posibilidades hay de acertar la longitud de la clave, y por tanto, de descifrar correctamente.

▼ Apartado (b)

Construir una tabla con las 8 letras de mayor frecuencia del criptograma.

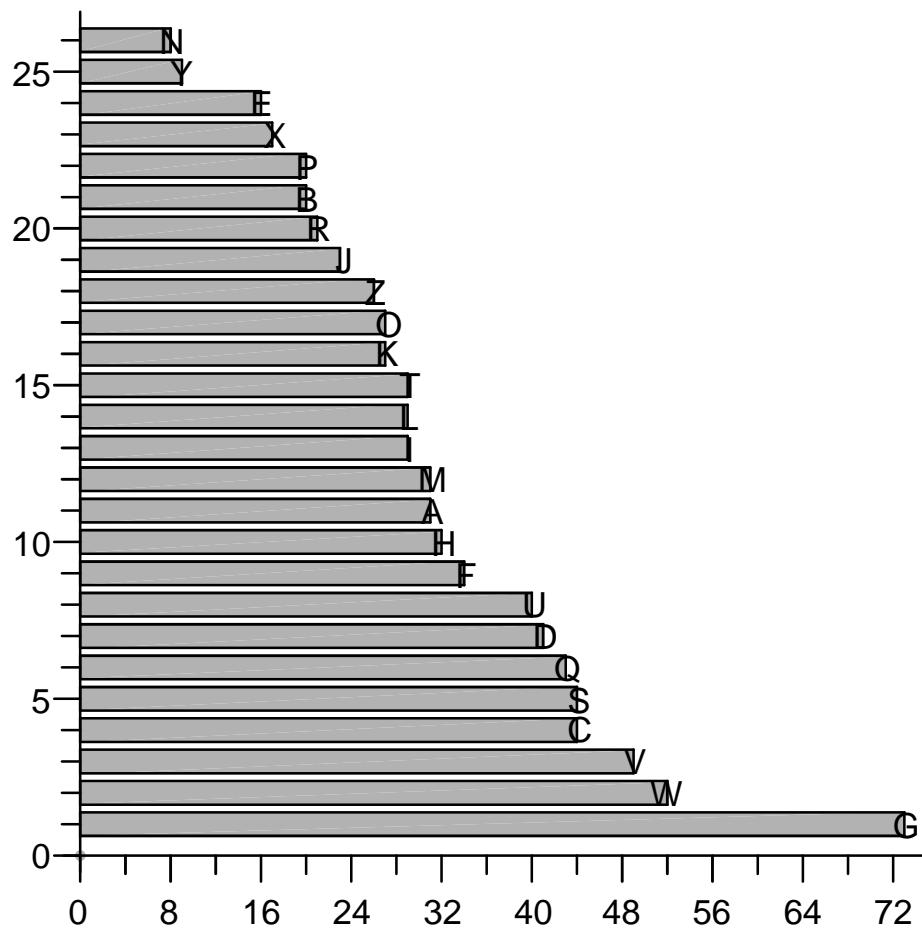
▼ Solución

Calculamos la frecuencia de repetición de letras en el criptograma (ordenadas de mayor a menor):

$$\begin{aligned}
 > \text{frecuencias} := \text{sort}([\text{CharacterFrequencies}(\text{criptograma}, \text{'upper'})], (u, v) \\
 &\quad \rightarrow \text{evalb}(\text{rhs}(u) \geq \text{rhs}(v))); \\
 \text{frecuencias} := [&"G" = 73, "W" = 52, "V" = 49, "C" = 44, "S" = 44, "Q" = 43, "D" = 41 \quad (3.1.1) \\
 &,"U" = 40, "F" = 34, "H" = 32, "A" = 31, "M" = 31, "I" = 29, "L" = 29, "T" = 29 \\
 &,"K" = 27, "O" = 27, "Z" = 26, "J" = 23, "R" = 21, "B" = 20, "P" = 20, "X" = 17 \\
 &,"E" = 16, "Y" = 9, "N" = 8]
 \end{aligned}$$

Hacemos una representación con diagrama de barras:

$$> \text{BarChart}(\text{frecuencias}, \text{color} = \text{aquamarine});$$



Recogemos las 8 más frecuentes en una tabla:

Letras con mayor frecuencia de repetición:	
"G" = 73	(3.1.2)
"W" = 52	(3.1.3)
"V" = 49	(3.1.4)
"C" = 44	(3.1.5)
"S" = 44	(3.1.6)
"Q" = 43	(3.1.7)
"D" = 41	(3.1.8)
"U" = 40	(3.1.9)

Apartado (c)

Averiguar si la palabra AAPEX aparece en el criptograma e indicar, si procede, el número de repeticiones y su posición.

Solución

Buscamos la palabra AAPEX:

```
> SearchAll("AAPEX", criptograma);  
267, 572  
(4.1.1)
```

Donde obtenemos dos posiciones en las que aparece. En total, AAPEX se repite dos veces, una en la posición 267, y otra en la posición 572.

Apartado (d)

Buscar cinco tetragramas repetidos y construidos con las letras de mayor frecuencia y localizar su posición.

Solución

Procedemos a localizar todos los grupos de cuatro letras (tetragramas) que se repitan en el criptograma (es decir, que existan dos veces al menos):

```
> buscarepetido := proc(texto, i, n)  
local tetragrama, posiciones;  
tetragrama := substring(texto, i..i + n - 1);  
posiciones := [SearchAll(tetragrama, texto)];  
if nops(posiciones) > 1 then RETURN(convert(cat(tetragrama), string));  
end if;  
end proc:  
> tetragramas := {seq(buscarepetido(criptograma, i, 4), i = 1 .. length(criptograma) - 1  
- 4)};  
tetragramas := {  
"GMJG", "IVQG", "HALC", "CZRA", "ZRAA", "RAAP", "AAPE", "APEX",  
"HOSD", "OSDO", "SOWH", "OWHV", "WHVL", "HVLQ", "PCVY", "CVYM"  
, "VYMG", "GGHX", "TSUI", "SUIE", "UIEQ", "IEQV", "EQVC", "QVCG",  
"GCZR", "TAPC"}  
> nops(%);  
26  
(5.1.2)
```

Existen 26 tetragramas repetidos, escogemos algunos de ellos y localizamos su posición: elegimos 5 que contengan las letras más frecuentes:

```
> cincotetragramas := ["GGHX", "IVQG", "GMJG", "GCZR", "QVCG"];  
cincotetragramas := ["GGHX", "IVQG", "GMJG", "GCZR", "QVCG"]  
(5.1.3)
```

```
> posiciones := seq(  
cincotetragramas[i], SearchAll(cincotetragramas[i], criptograma)], i = 1 .. 5);  
posiciones := [{"GGHX", 429, 584}, {"IVQG", 52, 627}, {"GMJG", 41, 216}  
, {"GCZR", 568, 643}, {"QVCG", 462, 517}]  
(5.1.4)
```

Apartado (e)

Calcular el n-grama repetido de mayor longitud y la diferencia entre sus posiciones

Solución

Ahora miramos el número de repeticiones de n-gramas, donde n va de 4 en adelante:
Escribimos una función que nos determine el número de n-gramas con repeticiones en función
de la longitud n.

```
> repeticiones := n → nops(
   {seq(buscarepetido(criptograma, i, n), i = 1 .. length(criptograma) - 1 - 4)}):
Probamos valores de n hasta obtener la mayor palabra con repetición:
> seq(repeticiones(n), n = 1 .. 10);
26, 207, 89, 26, 15, 10, 6, 3, 1, 0
```

(6.1.1)

Podemos ver que se cumple que hay 26 tetragramas repetidos, y un eneagrama repetido. Por tanto, $n=9$ (nueve letras). La mostramos:

```

> eneagrama := {seq(buscarepetido(criptograma, i, 9), i = 1 ..length(criptograma) - 1 - 4)};
                                         eneagrama := {"TSUIEQVCG"}                                     (6.1.2)

= > SearchAll(eneagrama[1], criptograma);
                                         457, 512                                     (6.1.3)

```

Vemos que está situado en las posiciones **457** y **512**, luego la diferencia entre repeticiones es **55**

Apartado (f)

Realizar una búsqueda exhaustiva para determinar las repeticiones de los trigramas y digramas, así como las posiciones que ocupan en el criptograma.

▼ Solución

Seguimos el mismo procedimiento que utilizamos para el enagrama del apartado anterior:

A continuación, el conjunto de trigramas y digramas repetidos:

```

    "OM", "JU", "NO" }

> trigramas := {seq(buscarepetido(criptograma, i, 3 ), i = 1 ..length(criptograma) - 1 -
4 )};

trigramas := {"FQG", "HAW", "JUI", "SUI", "ZWU", "TSU", "MKG", "KGB",      (7.1.2)
  "OWH", "GMJ", "GDU", "HCU", "RXG", "WDC", "CQD", "DAS", "MJG",
  "VVG", "IVQ", "VQG", "UGG", "HAL", "ALC", "JQG", "GHV", "CVI", "DUM",
  "UMU", "CGM", "MUQ", "TLK", "XGT", "TSO", "MFV", "TOQ", "OQB",
  "GGH", "VWF", "FVS", "DUW", "UWH", "CPD", "SCZ", "CZR", "ZRA",
  "RAA", "AAP", "APE", "PEX", "QDK", "GBH", "HOS", "OSD", "SDO", "RWD"
, "DLS", "WPC", "SOW", "WHV", "HVL", "VLQ", "QGD", "PCV", "CVY",
  "VYM", "YMG", "WTO", "AFO", "ETW", "YMK", "ZLV", "USG", "BHA",
  "DVV", "GHX", "OGW", "UIE", "IEQ", "EQV", "QVC", "VCG", "SIH", "MJQ",
  "QGP", "QBG", "GCZ", "TAP", "APC", "KSQ"}
```

Ahora escribimos en dos tablas la información de las posiciones para cada trígrama y digrama.

Diag

Dígramas repetidos hay **207** mientras que trigramas, **89** (los dígramas los repartimos en tres columnas de la spreadsheet).

```

> with(Spread):
> spreaddiramas := CreateSpreadsheet(Digramas):
```

Digram

A	B	C	D
1	"EC"	[164, 668, 784]	"QL"
2	"GW"	[436, 506]	"LM"
3	"WS"	[199, 683]	"MF"
4	"LV"	[386, 756]	"FV"
5	"BW"	[11, 172, 225, 340]	"VU"
6	"JG"	[43, 218, 468, 773]	"UA"
7	"VC"	[88, 154, 236, 463, 518, 606]	"VL"
8	"ZJ"	[72, 772]	"MT"
9	"GM"	[41, 101, 216, 231, 286]	"TO"
10	"MJ"	[42, 217, 552, 602, 722]	"OQ"
11	"WQ"	[341, 380]	"QB"
12	"JQ"	[73, 403, 553, 603]	"AG"
13	"BH"	[280, 420, 690]	"VW"
14	"FC"	[1, 84, 493, 614, 709]	"WF"
15	"CS"	[2, 765]	"SD"
16	"SP"	[3, 398, 735]	"MP"
17	"PR"	[4, 594]	"FW"
18	"RR"	[5, 646, 675]	"WI"
19	"RX"	[6, 131, 595]	"IW"
20	"XG"	[7, 132, 447, 617]	"DX"
21	"GU"	[8, 63, 243, 378, 388, 598]	"SF"
22	"US"	[9, 407, 639]	"QS"
23	"WD"	[12, 22, 182, 290, 415, 810]	"SC"
24	"DC"	[13, 23, 183, 191]	"NC"
25	"CQ"	[14, 439]	"MK"
26	"QD"	[15, 272, 440, 502, 800]	"VS"
27	"DA"	[16, 796]	"DQ"
28	"AS"	[17, 105, 180, 797]	"GL"

```
[> seq(SetCellFormula(Digramas, i, 1, digramas[i]), i=1..100);
[> seq(SetCellFormula(Digramas, i, 2, [SearchAll(digramas[i], criptograma)]), i=1..
100):
[> seq(SetCellFormula(Digramas, i, 3, digramas[i+100]), i=1..100);
[> seq(SetCellFormula(Digramas, i, 4, [SearchAll(digramas[i
+100], criptograma)]), i=1..100):
[> seq(SetCellFormula(Digramas, i, 5, digramas[i+200]), i=1..7);
[> seq(SetCellFormula(Digramas, i, 6, [SearchAll(digramas[i
+200], criptograma)]), i=1..7):
[> spreadtrigramas := CreateSpreadsheet(Trigramas):
```

Trigramas		
A	B	C
1	"FQG"	[706, 741]
2	"HAW"	[681, 746]
3	"JUI"	[723, 768]
4	"SUI"	[458, 513]
5	"ZWU"	[147, 787]
6	"TSU"	[457, 512]
7	"MKG"	[222, 392, 557]
8	"KGB"	[223, 278, 659]
9	"OWH"	[319, 649]
10	"GMJ"	[41, 216]
11	"GDU"	[219, 325]
12	"HCU"	[128, 349]
13	"RXG"	[6, 131]
14	"WDC"	[12, 22, 182]
15	"CQD"	[14, 439]
16	"DAS"	[16, 796]
17	"MJG"	[42, 217]
18	"VVG"	[47, 427]
19	"IVQ"	[52, 122, 627]
20	"VQG"	[53, 628]
21	"UGG"	[58, 238]
22	"HAL"	[66, 421]
23	"ALC"	[67, 422]
24	"JQG"	[73, 553]
25	"GHV"	[75, 560]
26	"CVI"	[80, 310]
27	"DUM"	[95, 220]
28	"UMU"	[96, 496]

```

    > seq(SetCellFormula(Trigramas, i, 1, trigramas[i]), i = 1 ..nops(trigramas));
    > seq(SetCellFormula(Trigramas, i, 2, [SearchAll(trigramas[i], criptograma)]), i = 1 ..
           nops(trigramas)) :

```

▼ Apartado (g)

Hallar la longitud de la clave a partir de las posiciones de repetición de los tetragramas. ¿Es este resultado compatible con cualquier n-grama ($n=2,3,\dots$)?

▼ Solución

Analizamos la distancia de repetición para los 26 tetragramas obtenidos:

Creamos un proceso distancia que calcula la distancia de repetición dado un n-grama.

```

> distancia := proc(ngrama)
local p;
p := SearchAll(ngrama, criptograma);
RETURN(p[2] - p[1]);
end proc;

> distancias := [seq(distancia(tetragramas[i]), i = 1 ..nops(tetragramas))];
distancias := [175, 575, 355, 305, 305, 305, 305, 305, 205, 205, 330, 330, 330, 330,
               250, 250, 250, 155, 55, 55, 55, 55, 55, 55, 75, 185] (8.1.1)

```

Calculamos el máximo común divisor de entre estas distancias, cogiendo de dos en dos:

```

> divisores := [seq(seq(gcd(distancias[i], distancias[j])), i = 1 ..nops(distancias)), j = 1 ..
           nops(distancias))] :

```

Donde obtenemos que el máximo común divisor de las distancias de repeticiones es **5**. Luego la **longitud de la clave es 5**.

```
> longitudclave := 5 :
```

El resultado en general no será compatible, sino que se darán algunas coincidencias. Cuanto mayor sea n , mejor obtendremos la verdadera longitud de la clave (mejor descifraremos).

Puede darse el caso que obtengamos el mismo tamaño de clave para un n menor, veamos qué ocurre:

▼ Compatibilidad con digramas:

```

> distanciasdi := [seq(distancia(digramas[i]), i = 1 ..nops(digramas))];
distanciasdi := [504, 70, 484, 370, 161, 175, 66, 700, 60, 175, 39, 330, 140, 83, (8.1.1.1)
               763, 395, 590, 641, 125, 125, 55, 398, 10, 10, 425, 257, 780, 88, 62, 56, 415
               , 335, 130, 143, 745, 20, 15, 190, 1, 649, 45, 30, 380, 659, 244, 70, 20, 276,
               40, 180, 159, 230, 410, 26, 355, 355, 455, 450, 100, 254, 292, 80, 650,
               99, 612, 25, 125, 445, 155, 115, 46, 305, 375, 122, 430, 465, 413, 413, 303,
               106, 676, 40, 608, 81, 15, 120, 221, 108, 315, 3, 3, 143, 175, 295, 590, 36,
               613, 155, 210, 166, 16, 145, 32, 406, 195, 156, 640, 175, 175, 248, 509, 347
               , 8, 10, 505, 295, 29, 520, 200, 88, 190, 55, 95, 65, 70, 498, 169, 545, 81,
               385, 55, 165, 165, 529, 385, 105, 275, 10, 305, 305, 305, 120, 143, 65, 205,
               160, 150, 125, 25, 255, 180, 25, 25, 330, 330, 250, 307, 339, 335, 456, 90,
               290, 137, 304, 10, 355, 192, 41, 243, 240, 83, 80, 300, 370, 196, 174, 80,
               180, 52, 50, 155, 66, 40, 55, 55, 55, 310, 295, 26, 45, 290, 260, 170, 37, 100
               , 38, 156, 208, 185, 109, 115, 28, 65, 84, 45, 10]
> divisoresdi := [seq(seq(gcd(distanciasdi[i], distanciasdi[j])), i = 1 ..nops(distanciasdi)), j = 1 ..nops(distanciasdi)] :

```

$\dots nops(distanciasdi)), j = 1 \dots nops(distanciasdi))]$:
El máximo común divisor es 1 y por tanto la longitud de la clave es 1. **Erróneo.**

Compatibilidad con trigramas:

```
> distanciastri := [seq(distancia(trigramas[i]), i = 1 ..nops(trigramas))];  
distanciastri := [35, 65, 45, 55, 640, 55, 170, 55, 330, 175, 106, 221, 125, 10,  
425, 780, 175, 380, 70, 575, 180, 355, 355, 480, 485, 230, 125, 400, 115,  
395, 413, 315, 3, 145, 175, 395, 255, 347, 70, 81, 343, 165, 490, 305, 275,  
305, 305, 305, 230, 410, 205, 205, 160, 125, 460, 25, 330, 330, 330,  
330, 305, 250, 250, 250, 90, 10, 355, 250, 370, 232, 270, 50, 155, 70,  
55, 55, 55, 55, 170, 50, 188, 215, 75, 185, 185, 145 ]
```

```
> divisorestri := [seq(seq(gcd(distanciastri[i], distanciastri[j])), i = 1  
..nops(distanciastri)), j = 1 ..nops(distanciastri))]
```

El máximo común divisor es 1 y por tanto la longitud de la clave es 1. **Erróneo.**

Apartado (h)

Crear las listas de caracteres -tantas como la longitud de la clave- asociadas a los análisis elementales de cífrados de transposición.

Solución

Generamos n-gramas con la longitud de la clave, es decir, 5 (troceamos el criptograma en palabras de cinco letras).

```
> pentagramas := [seq(convert(cat(seq(criptograma[i  
+ k], k = 0 ..longitudclave - 1)), string), i = seq(5j  
+ 1, j = 0 .. length(criptograma) - 1))];  
pentagramas := [
```

```
"FCSPR", "RXGUS", "BWDCQ", "DASVI", "YWDCV", "DJANW", "GIVGB",  
"XVEGG", "GMJGJ", "HVVGF", "OIVQG", "YMUGG", "DLGUH", "HALCT",  
"HZJQG", "HVYTC", "VIFFC", "FIVCJ", "HHWND", "UMUKC", "GMUQA",  
"SZSGZ", "XTLKA", "RKGOD", "UIVQF", "QWHCU", "RXGTS", "OTSOS",  
"QWKFS", "WZWUQ", "LMFVC", "VUANT", "UIFEC", "VLMTO",  
"QBWGG", "HBAGA", "SWDCF", "VWFPS", "DCMPW", "FWIWS", "DXSTS"  
, "FQSCH", "LBMNC", "GMJGD", "UMKGB", "WIFVS", "GMDQG",  
"VCUGG", "LDGUD", "UWHKS", "WIJKC", "VBJCP", "DRSCZ", "RAAPE",  
"XQDKB", "RAKGB", "HOSDO", "GMKRW", "DLSFO", "PMFVS", "DZWPC"  
, "YIJNC", "VIJTS", "QLSOW", "HVLQG", "DUWPC", "VYMGQ", "RVKKB",  
"WQWTO", "QMFHC", "UUAFO", "ETWUG", "XJAFO", "VLWCZ",  
"TCANS", "UTGUW", "QYMKZ", "LVGUE", "XMKGC", "OQSPZ", "DXJQL",  
"LUSGL", "SZGRW", "DKAQB", "HALCP", "DVVGG", "HXWTO",  
"GWKCQ", "DJSDO", "QXGTO", "FMHVO", "UTSUI", "EQVCG", "RJJGH",  
"RLGEI", "DVVQZ", "DZSIF", "HOSDO", "FWFCW", "UMUQB", "FQDKO",  
"GWJSI", "HTSUI", "EQVCG", "HZACT", "LKLKQ", "LIVWF", "DVLGZ",  
"RAUKB", "FWHTW", "PMJQG", "PMKGG", "HUVUWO", "QBG CZ",  
"RAAPE", "XTAPC", "VYMGG", "HXMUW", "HZGPR", "XZGUT", "XMJQB"  
, "VCKVW", "WCAFC", "VXGTD", "DVACU", "XIVQG", "DYMKS",
```

```

"QMKUS", "GQGCZ", "RRSOW", "HVLQU", "UILKG", "BYMGT", "LZECF",
"RVLQR", "RTGSI", "HAWSI", "LAGGB", "HAGJI", "EWMPP", "HWWHW",
"FGFC", "ETWGZ", "DTIWW", "OMJUI", "EQSAZ", "DQFFS", "PVABO",
"FGPF", "HAWTJ", "DLSCZ", "LVIWZ", "LTAPC", "SWJUI", "DZJGB",
"GIEKS", "QBGE", "UZWUD", "RVVKO", "DASEQ", "DZVNO", "MIMTW",
"DHGNO" ]

```

Generamos las listas del primer caracter de las palabras, segundo caracter... así hasta el la lista del quinto caracter.

```

> lista := n → [seq(substring(pentagramas[i], n ..n), i = 1 ..nops(pentagramas))]: (9.1.2)
> lista(1);
[F", "R", "B", "D", "Y", "D", "G", "X", "G", "H", "O", "Y", "D", "H", "H", "H", "V",
 "F", "H", "U", "G", "S", "X", "R", "U", "Q", "R", "O", "Q", "W", "L", "V", "U",
 "V", "Q", "H", "S", "V", "D", "F", "D", "F", "L", "G", "U", "W", "G", "V", "L",
 "U", "W", "V", "D", "R", "X", "R", "H", "G", "D", "P", "D", "Y", "V", "Q", "H",
 "D", "V", "R", "W", "Q", "U", "E", "X", "V", "T", "U", "Q", "L", "X", "O", "D",
 "L", "S", "D", "H", "D", "G", "D", "Q", "F", "U", "E", "R", "R", "D", "D", "H"
 , "F", "U", "F", "G", "H", "E", "H", "L", "L", "D", "R", "F", "P", "P", "H", "Q", "R"
 , "X", "V", "H", "H", "X", "X", "V", "W", "V", "D", "X", "D", "Q", "G", "R", "H",
 "U", "B", "L", "R", "R", "H", "L", "H", "E", "H", "F", "E", "D", "O", "E", "D", "P",
 "F", "H", "D", "L", "L", "S", "D", "G", "Q", "U", "R", "D", "D", "M", "D"]
> lista(2);
[C", "X", "W", "A", "W", "J", "I", "V", "M", "V", "I", "M", "L", "A", "Z", "V", "I", "I"
 , "H", "M", "M", "Z", "T", "K", "I", "W", "X", "T", "W", "Z", "M", "U", "I", "L",
 "B", "B", "W", "W", "C", "W", "X", "Q", "B", "M", "M", "I", "M", "C", "D", "W",
 "I", "B", "R", "A", "Q", "A", "O", "M", "L", "M", "Z", "I", "I", "L", "V", "U", "Y",
 "V", "Q", "M", "U", "T", "J", "L", "C", "T", "Y", "V", "M", "Q", "X", "U", "Z",
 "K", "A", "V", "X", "W", "J", "X", "M", "T", "Q", "J", "L", "V", "Z", "O", "W",
 "M", "Q", "W", "T", "Q", "Z", "K", "I", "V", "A", "W", "M", "M", "V", "B", "A",
 "T", "Y", "X", "Z", "Z", "M", "C", "C", "X", "V", "I", "Y", "M", "Q", "R", "V", "I",
 "Y", "Z", "V", "T", "A", "A", "W", "V", "Q", "T", "M", "Q", "V",
 "Q", "A", "L", "V", "T", "W", "Z", "I", "B", "Z", "V", "A", "Z", "I", "H"] (9.1.3)
> lista(3);
[S", "G", "D", "S", "D", "A", "V", "E", "J", "V", "V", "U", "G", "L", "J", "Y", "F",
 "V", "W", "U", "U", "S", "L", "G", "V", "H", "G", "S", "K", "W", "F", "A", "F",
 "M", "W", "A", "D", "F", "M", "I", "S", "S", "M", "J", "K", "F", "D", "U", "G", "H"
 , "J", "J", "S", "A", "D", "K", "S", "K", "S", "F", "W", "J", "J", "S", "L", "W", "M",
 "K", "W", "F", "A", "W", "A", "W", "A", "G", "M", "G", "K", "S", "J", "S", "G",
 "A", "L", "V", "W", "K", "S", "G", "H", "S", "V", "J", "G", "V", "S", "S", "F", "U",
 "D", "J", "S", "V", "A", "L", "V", "L", "U", "H", "J", "K", "U", "G", "A", "A", "M"
 , "M", "G", "G", "J", "K", "A", "G", "A", "V", "M", "K", "G", "S", "L", "L", "M",
 "E", "L", "G", "W", "G", "G", "M", "W", "G", "W", "I", "J", "S", "F", "A", "G",
 "W", "S", "I", "A", "J", "J", "E", "G", "W", "V", "S", "V", "M", "G"] (9.1.4)
> lista(4);
[P", "U", "C", "V", "C", "N", "G", "G", "G", "Q", "G", "G", "U", "C", "Q", "T", "F",
 "C", "N", "K", "Q", "G", "K", "O", "Q", "C", "T", "O", "F", "U", "V", "N", "E", "T"] (9.1.5)

```

```

, "G", "G", "C", "P", "P", "W", "T", "C", "N", "G", "G", "V", "Q", "G", "U", "K",
"K", "C", "C", "P", "K", "G", "D", "R", "F", "V", "P", "N", "T", "O", "Q", "P", "G",
"K", "T", "H", "F", "U", "F", "C", "N", "U", "K", "U", "G", "P", "Q", "G", "R", "Q"
, "C", "G", "T", "C", "D", "T", "V", "U", "C", "G", "E", "Q", "I", "D", "C", "Q", "K"
, "S", "U", "C", "C", "K", "W", "G", "K", "T", "Q", "G", "W", "C", "P", "P", "G",
"U", "P", "U", "Q", "V", "F", "T", "C", "Q", "K", "U", "C", "O", "Q", "K", "G", "C"
, "Q", "S", "S", "G", "J", "P", "H", "F", "G", "W", "U", "A", "F", "B", "P", "T", "C"
, "W", "P", "U", "G", "K", "E", "U", "K", "E", "N", "T", "N"]

```

De esta manera conseguimos reducir el problema a 5 más simples que siguen un cifrado CAESAR.

▼ Apartado (i)

Representar los diagramas de barras frecuenciales de cada lista y deducir de ellos la clave monoalfabética de transposición.

▼ Solución

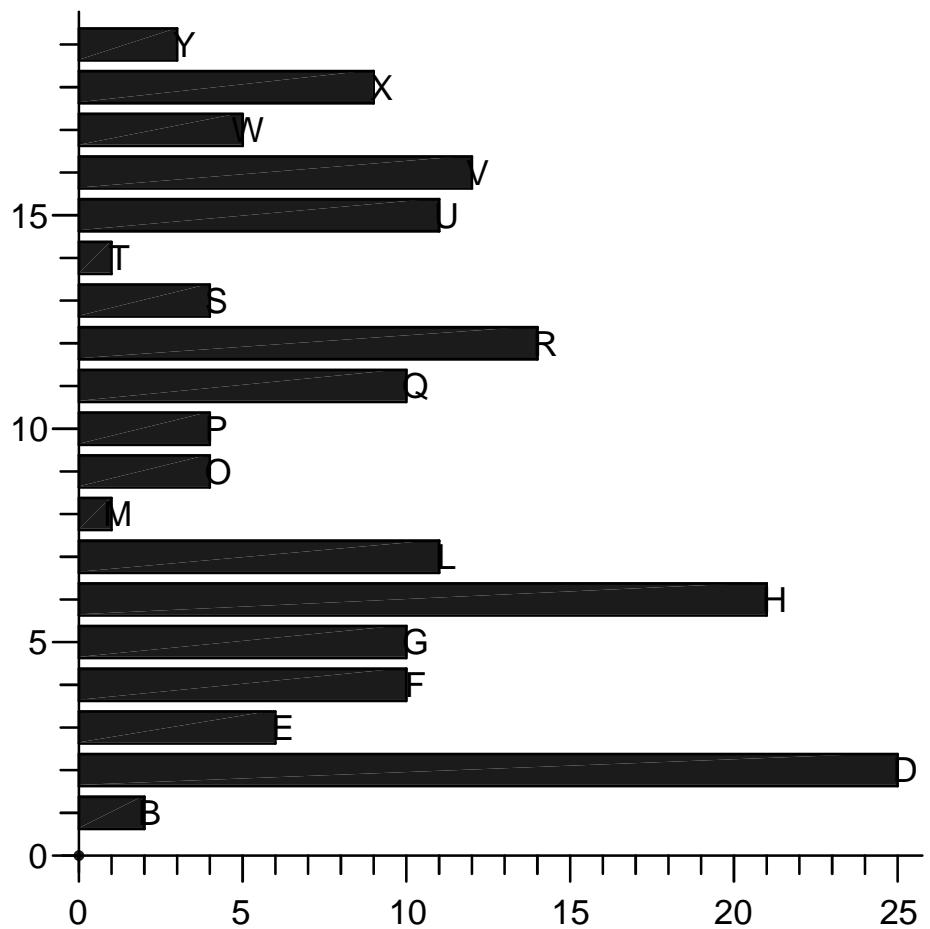
Concatenamos las letras de las listas 1 a 5:

```

> freccaracteres := n → [
  CharacterFrequencies(
    convert(cat(seq(lista(n)[i], i = 1 .. nops(lista(n)))), string), 'upper')
  ) :
  Representamos la frecuencia de aparición de letras en histogramas para cada carácter.

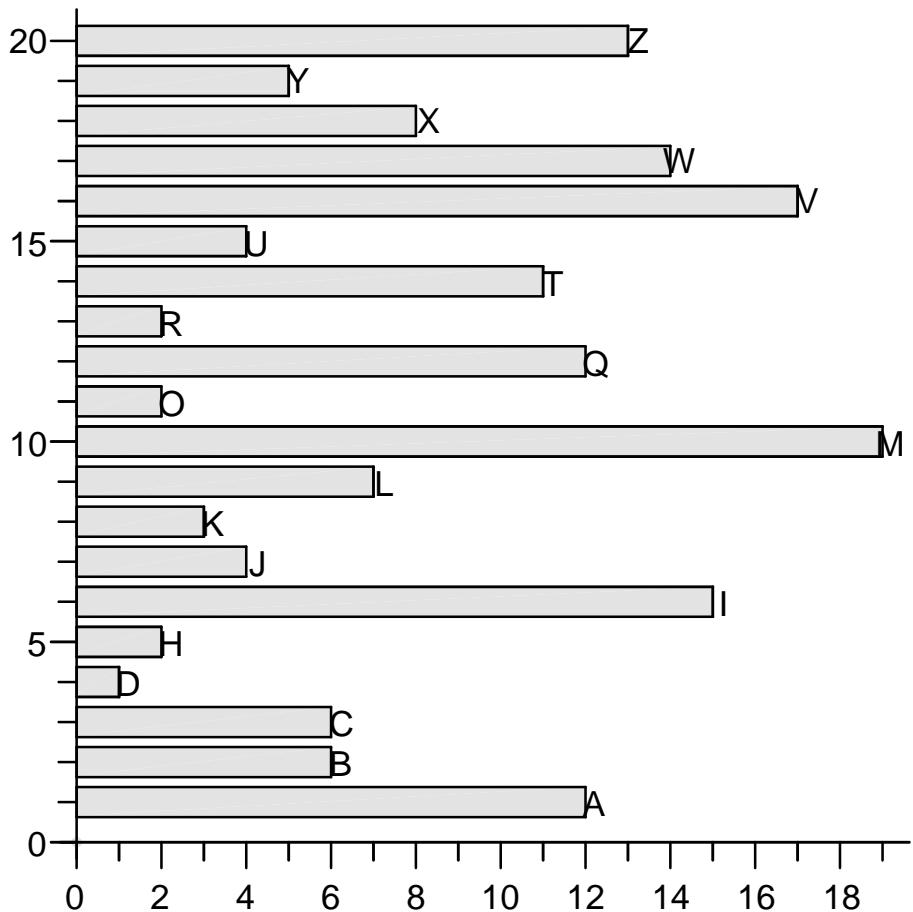
```

```
> BarChart(freccaracteres(1), color = blue);
```



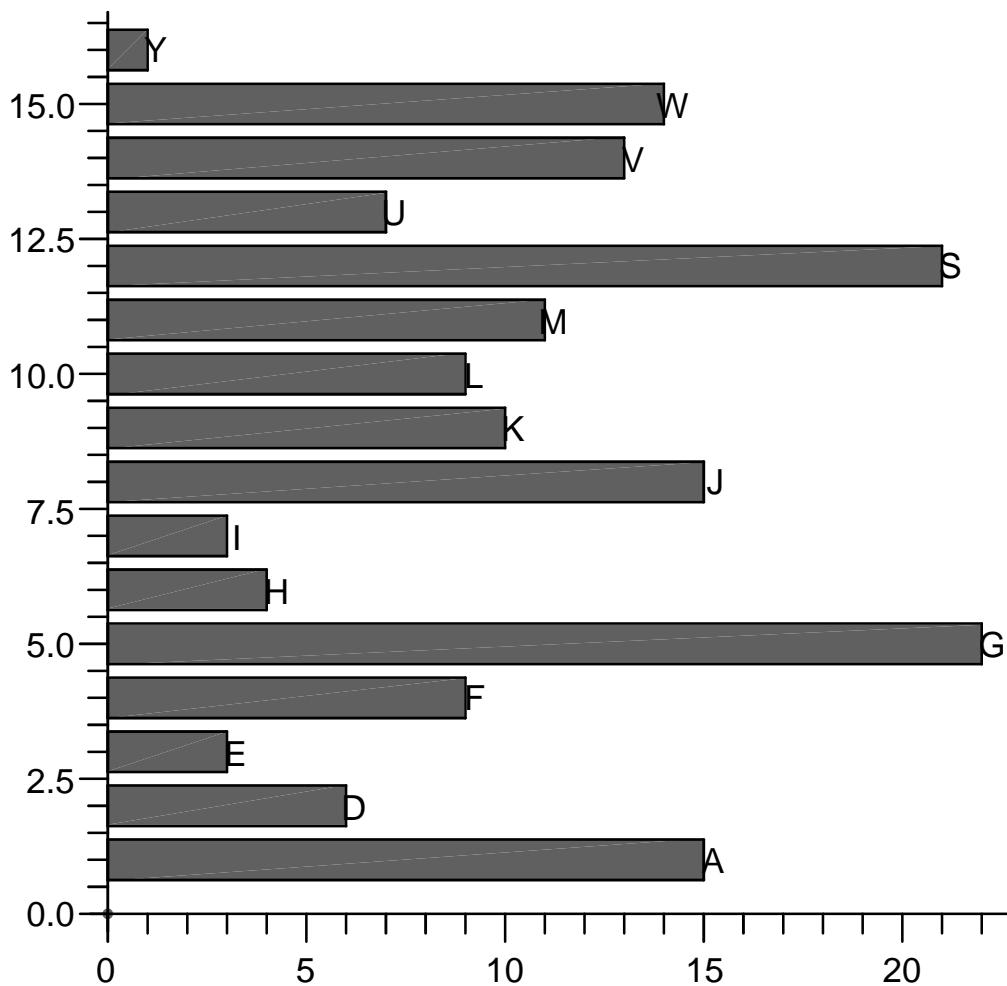
Vemos que la letra más frecuente para la posición 1 es la **D**.

> *BarChart(freccaracteres(2), color=yellow);*



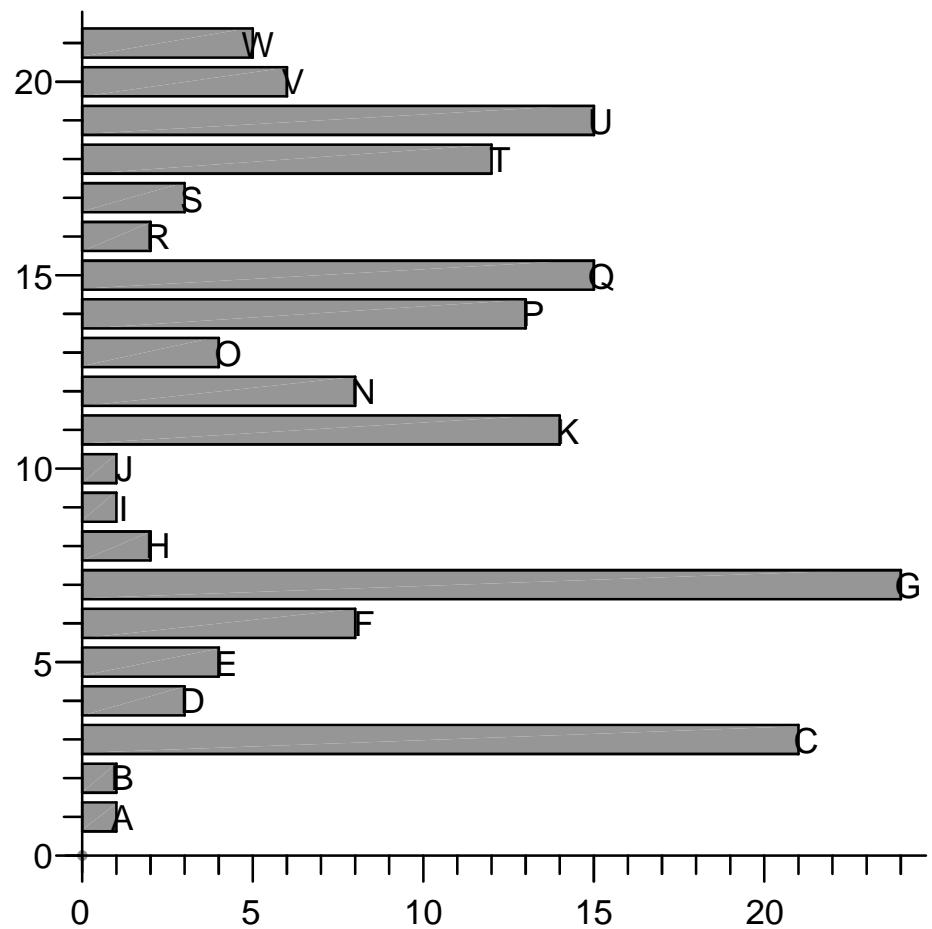
Vemos que la letra más frecuente para la posición 2 es la **M**.

```
> BarChart(freccaracteres(3 ), color=orange);
```



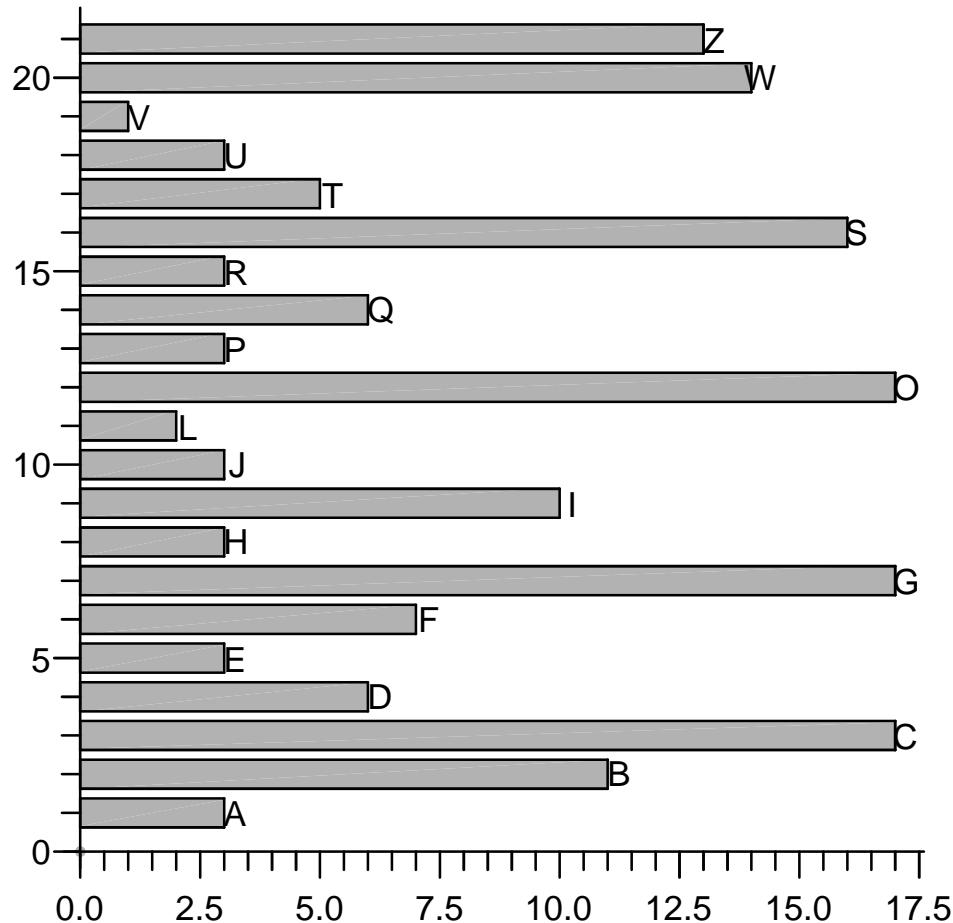
Vemos que la letra más frecuente para la posición 3 es la **G**.

> *BarChart(freccaracteres(4), color = green);*



Vemos que la letra más frecuente para la posición 4 es la **G**.

> *BarChart(freccaracteres(5), color = aquamarine);*



Vemos que esta vez hay tres letras que aparecen con más frecuencia para la posición 5: **C, G, O y S.**

Tenemos las 4 primeras letras de la clave cifrada, pero la última puede ser 1 entre las 4 posibilidades. Es decir:

[> *clavescifradas :=["DMGGC", "DMGGG", "DMGGO", "DMGGS"];*

▼ Apartado (j)

Especificar la clave de cifrado, descifrar el criptograma y comprobar que el mensaje claro es legible.

▼ Solución

Las letras más probables en español son: A, E, O y S (Fuente: *Wikipedia en Español, artículo: Letras más frecuentes*).

[> *abecedario := proc(i)*

RETURN(substring(ABCDEFGHIJKLMNOPQRSTUVWXYZ, i..i));
end proc :

Escribimos el procedimiento para obtener la clave a partir de un mensaje y su encriptado:

[> *obtenerclave := proc(mensaje, cifrado)*

```

local i, c1, c2, c3;
clave := cat(seq("A", i = 1 ..length(mensaje)));
for i from 1 to length(mensaje) do
c1 := convert(mensaje[i],'bytes')[1];
c2 := convert(cifrado[i],'bytes')[1];
clave[i] := abecedario((c2 - c1 mod 26) + 1);
end do;
clavefinal := cat(seq(clave[i], i = 1 ..length(mensaje)));
end proc:

```

Ahora obtenemos la clave a partir de combinaciones de las letras más frecuentes del castellano, A, E, O y S.

```

> posiblesclaves := x → [seq(obtenerclave(x, clavescifradas[i]), i = 1 ..4)] :
> posiblesclaves ("AAAAAA");
[DMGGC, DMGGG, DMGGO, DMGGS] (11.1.1)
> posiblesclaves ("EEEEEE");
[ZICCY, ZICCC, ZICCK, ZICCO] (11.1.2)
> posiblesclaves ("OOOOO");
[PYSSO, PYSSS, PYSSA, PYSSE] (11.1.3)
> posiblesclaves ("SSSSS");
[LUOOK, LUOOO, LUOOW, LUOOA] (11.1.4)
> posiblesclaves ("AEOSA");
[DISOC, DISOG, DISOO, DISOS] (11.1.5)
> posiblesclaves ("AEOAE");
[DISGY, DISGC, DISGK, DISGO] (11.1.6)
> posiblesclaves ("AEOEA");
[DISCC, DISCG, DISCO, DISCS] (11.1.7)

```

Vemos que la clave legible de todas las posibles que hemos buscado es la palabra **DISCO**.

Ahora vamos a utilizar esta clave para descifrar el texto.

```

> caesar := proc(letra, clave)
local x;
x := letra;
if 64 < x then
if x < 91 then
x := 65 + (mod(x - 65 + clave, 26))
end if;
end if;
x
end proc:
> cifrar := proc(mensaje, clave)
local i, j, criptobytes, clavebytes;
criptobytes := convert(mensaje,'bytes');
clavebytes := convert(clave,'bytes');
for i from 0 to (length(mensaje) - 1) do
for j from 1 to length(clave) do
criptobytes[i · length(clave) + j] := map(caesar, criptobytes[i · length(clave) + j], (clavebytes[j] - 65)):
end do;
end do;
print(convert(criptobytes,'bytes'))

```

```

    end proc :

> descifrar := proc(cripto, clave)
local i,j, criptobytes, clavebytes;
criptobytes := convert(cripto,'bytes');
clavebytes := convert(clave,'bytes');
for i from 0 to  $\left(\frac{\text{length}(\text{cripto})}{\text{length}(\text{clave})} - 1\right)$  do
for j from 1 to length(clave) do
criptobytes[i · length(clave) + j] := map(caesar, criptobytes[i · length(clave) + j], -(clavebytes[j] - 65)):
end do;
end do;
print(convert(criptobytes,'bytes'))
end proc :

```

Ya que tenemos los métodos para cifrar y descifrar un mensaje, vamos a descomponer el criptograma en varias partes de 50 letras (líneas del texto original) para ir descifrándolo:

```

> for i from 1 to 16 do
cripto[i] := substring(criptograma, ((i - 1) · 50) + 1 .. 50 · i);
end do ;
> cripto[17] := substring(criptograma, 801 .. 815) :
> seq(descifrar(cripto[i], "DISCO"), i = 1 .. 17);
"CUANDOPOSEYOLACASATUVOLA HABILIDADENUNMESDREVENDER"
" LADOSVECESADOSTESTAFERROSENGROSANDOCADAVEZELPRECIO"
"DECOMPRAEULTIMOCOMPRADORNOPAGOPORELLAMENOSDETRESC \
"
"IENTOSMILFRANCOSDURANTEESETIEMPOLARSONNEAUUNICOQUE"
"APARECIAATITULODEREPRESENTANTEDELOSSUCESIVOSPROPIE"
" TARIOSTRABAJAALOSINQUILONESNEGABADESPIADADAMENTE"
" ARENOVARLOSARRENDAMIENTOSAMENOSQUECONSINTIERANENFO"
" RMIDABLESSUBIDASDEALQUIERLOSINQUILONOSQUESEOLIANL"
"APROXIMAEXPROPIACIONESTABANDESEPERADOSACABABANPORA"
"CEPTARLASUBIDASOBRETODOCUANDOLARAGREGABA CONAIRECON"
" CILIADORQUELASUBIDASERIAFICTICIADURANTELOS CINCOPRI"
" MEROSEMESESENCUANTOALOSINQUILONOSQUESEPUISIERONDUROS"
" UERONSUSTITUIDOSPORPANIAGUADOSAQUIENESSEDIOALOJAMI"
" ENTOGRATISYQUEFIRMARONTODOLOQUESEQUISOENESOHUBOUNB"
" ENEFICIODOBLEELALQUIERSUBIAYLAINDEMNZACIONRESERV"
" ADAALINQUILINOPORSUARRENDAMIENTOCORRESPONDIAASACC"
" ARDLAJAURIAZOLA"

```

(11.1.8)

Vemos que el mensaje es legible, pero sin tildes ni signos de puntuación.

Apartado (k)

Escribir el mensaje claro con los signos de puntuación correspondientes. ¿Quién es el autor de este fragmento?

▼ Solución

El texto reescrito es:

"Cuando poseyó la casa, tuvo la habilidad en un mes de revenderla dos veces a dos testaferros engrosando cada vez el precio de compra. El último comprador no pagó por ella menos de trescientosmil francos. Durante ese tiempo, Larsonneau, único que aparecía a título de representante de los sucesivos propietarios, trabaja a los inquilinos. Se negaba despiadadamente a renovar los arrendamientos a menos que consintieran en formidables subidas de alquiler. Los inquilinos, que se oían la próxima expropiación, estaban desesperados. Acababan por aceptar la subida. Sobre todo, cuando Lar agregaba con aire conciliador, que la subida sería ficticia durante los cinco primeros meses. En cuanto a los inquilinos que se pusieron duros, fueron sustituidos por paniaguados, a quienes se dio alojamiento gratis y que firmaron todo lo que se quiso. En eso hubo un beneficio doble: el alquiler subía y la indemnización reservada al inquilino por su arrendamiento correspondía a Saccard. La Jauría. Zola"

Como se puede leer al final del texto, este fragmento pertenece al libro **La Jauría**, de **Émile Zola**.

▼ Apartado (I)

Cifrar el mensaje claro con todos sus detalles sintácticos (signos de puntuación, acentos, mayúsculas/minúsculas, etc.) utilizando codificación ASCII.

▼ Solución

Tenemos el fragmento de texto que queremos cifrar en ASCII:

```
> fragmento := "Cuando poseyó la casa, tuvo la habilidad en un mes de revenderla  
dos veces a dos testaferros engrosando cada vez el precio de compra. El último  
comprador no pagó por ella menos de trescientosmil francos. Durante ese tiempo  
, Larsonneau, único que aparecía a título de representante de los sucesivos  
propietarios, trabaja a los inquilinos. Se negaba despiadadamente a renovar los  
arrendamientos a menos que consintieran en formidables subidas de alquiler.  
Los inquilinos, que se oían la próxima expropiación, estaban desesperados.  
Acababan por aceptar la subida. Sobre todo, cuando Lar agregaba con aire  
conciliador, que la subida sería ficticia durante los cinco primeros meses. En  
cuanto a los inquilinos que se pusieron duros, fueron sustituidos por  
paniguados, a quienes se dio alojamiento gratis y que firmaron todo lo que se  
quiso. En eso hubo un beneficio doble: el alquiler subía y la indemnización  
reservada al inquilino por su arrendamiento correspondía a Saccard. La Jauría.  
Zola":  
> ASCII := proc(letra, clave)  
local x, y;  
x := letra;  
x := (x + clave)mod 255;  
x;  
end proc:  
> cifrarASCII := proc(mensaje, clave)  
local i, j, criptobytes, clavebytes;  
criptobytes := convert(mensaje,'bytes');  
clavebytes := convert(clave,'bytes');  
for i from 0 to  $\left(\frac{\text{length}(\text{mensaje})}{\text{length}(\text{clave})} - 1\right)$  do  
for j from 1 to length(clave) do  
criptobytes[i · length(clave) + j] := map(ASCII, criptobytes[i · length(clave) + j], clave);  
end do;  
end do;
```

```

+j], (clavebytes[j] - 65)) :
end do;
end do;
print(convert(criptobytes,'bytes'))
end proc:
> descifrarASCII:=proc(mensaje, clave)
local i,j, criptobytes, clavebytes;
criptobytes := convert(mensaje,'bytes');
clavebytes := convert(clave,'bytes');
for i from 0 to  $\left(\frac{\text{length}(\text{mensaje})}{\text{length}(\text{clave})} - 1\right)$  do
for j from 1 to length(clave) do
criptobytes[i · length(clave) + j] := map(ASCII, criptobytes[i · length(clave)
+j], - (clavebytes[j] - 65)):
end do;
end do;
print(convert(criptobytes,'bytes'))
end proc:
> longitud := length(fragmento);
longitud := 1095

```

(13.1.1)

Ahora separamos el fragmento de texto en varias partes y las vamos encriptando en ASCII, como ya hicimos para el descifrado del mensaje sin signos de puntuación.

```

> for i from 1 to 21 do
linea[i] := substring(fragmento, ((i - 1) · 50) + 1 .. 50 · i);
end do;
linea1 := "Cuando poseyó la casa, tuvo la habilidad en"
linea2 := " un mes de revenderla dos veces a dos testaferros "
linea3 := "engrosando cada vez el precio de compra. El &uacut"
linea4 := "elñltimo comprador no pagó por ella menos de"
linea5 := " trescientosmil francos. Durante ese tiempo, Larso"
linea6 := "nneau, único que aparecía a t&iacute"
linea7 := ";tulo de representante de los sucesivos propietari"
linea8 := "os, trabaja a los inquilinos. Se negaba despiadada"
linea9 := "mente a renovar los arrendamientos a menos que con"
linea10 := "sintieran en formidables subidas de alquiler. Los "
linea11 := "inquilinos, que se olían la próxima "
linea12 := "expropiación, estaban desesperados. Acababa"
linea13 := "n por aceptar la subida. Sobre todo, cuando Lar ag"
linea14 := "regaba con aire conciliador, que la subida ser&iac"
linea15 := "ute;a ficticia durante los cinco primeros meses. E"
linea16 := "n cuanto a los inquilinos que se pusieron duros, f"
linea17 := "ueron sustituídos por paniaguados, a quiene"
linea18 := "s se dio alojamiento gratis y que firmaron todo lo"
linea19 := " que se quiso. En eso hubo un beneficio doble: el "

```

*linea*₂₀ := "alquiler subía y la indemnización re"

*linea*₂₁ := "servada al inquilino por su arrendamiento correspo" (13.1.2)

> *linea*[22] := *substring*(*fragmento*, 1051 ..1095);

*linea*₂₂ := "ndía a Saccard. La Jauría. Zola" (13.1.3)

> *seq*(*cifrarASCII*(*linea*[*i*], "DISCO"), *i* = 1 ..22);

"F}sprr(,q□h□8qof}†gI#ts"qd{..w}^q.oi2joeq~krdl2g|"
"#}€" {h{2fs#zwxsqlwtzd(vq□#~wesv(s"rr{2vsv|shsuz□u."
"hvvt}vi€f}#ksfo#~w|.ht2r€hk{q.gm2e}px,,c<#M~"4xiuw,"
"hC~vwpw2e}px,,crrz2p}#xsi4riuw,hC2r}u(wnzd(□g|r{2fs"
"#|,g□fqwp,r{□kz#n,,c|fw...0.G},,c|wm2g□h(†kspx□..Oi,,u}"
"qvwc/(8wof}†gIqquq.t}w"osi,,gq)qsefwmMc.d(†(wdk‡vs"
">|‡n}#lw"€hx,,g□hv†c|wm2fs#t□u.v}ug□l~□u.sz□rwh|stw"
"r{>",uitcxd(s"zr{2k|t}nwqw...0.Vm2psjtc.gm...rwdsfo"
"pm€vs#i2tsqw^c€#t□u.dz,,g|gi□ksq|□u.d(□g|r{2sfh(uq|"
"vq€vwhzsp.hv2h}uu{foetu.v}tkrd{2fs#i~sfltwt<#T□u."
"lvfwwoq€q□/(fws#{w"}o.{cqx|w=oq(~c.sz8qof}†gI{q□c."
"h€,t}sqsew)wsefwmMp:#m...voei"rh{wu~hzsf}v62Cqdjsdo"
"q(,q€#iug~wi,,zd(...wplls0.Vwtts#|□f)/(uwoql□"Zdz2cu"
"umycpd(uq|#i{ts#k□pqlt{crrz>"□xm2no#{‡dwgi2usu.{cq"
"x|w=o#n{e,lk{c.g},,c|wm2n}v(uk|fw2r€luwt}v(□g□h{@"S"
"q(uwoq|□"o#t□u.lvfwwoq€q□#y‡g.vm2rfvqwt}q(vw€r{>"t"
"xm,,q|#{‡u,l|‡(wdk‡vs>1□u.sw,,~dv{cuxivq□/(s"□xqwp"
"v(...g.gq□"oow|c{lm€v}#o,,c,l{2{.t}w"tlz□c€rv2v}gw2n}"
"#y‡g.vm2sf1{□0.Hv2g□r(zwpr(‡p.em€gtlk{q.gwtns=(wn."
"dtfwwom,"□xj8kof}†gId(<"zd({prhu€k^dk{()dk‡vs>v2ts"
"vm,,xogi2cz#q€sflt{p}#x□t.v}2c€um€fopqwp,r(uq€um...r}"
"ql8kof}†gId(s"adkuc€g62No#Rsw€)qsefwmMc<#b□no" (13.1.4)

Y con ello queda cifrado en ASCII el fragmento de texto de **La Jauría** con sus signos de puntuación.

Función Seno Cardinal

Ruben Darias Bello - Luciano Rubio Romero

Sistemas de Cálculo Simbólico

Grupo 0416, Curso 2007/08

▼ Enunciado

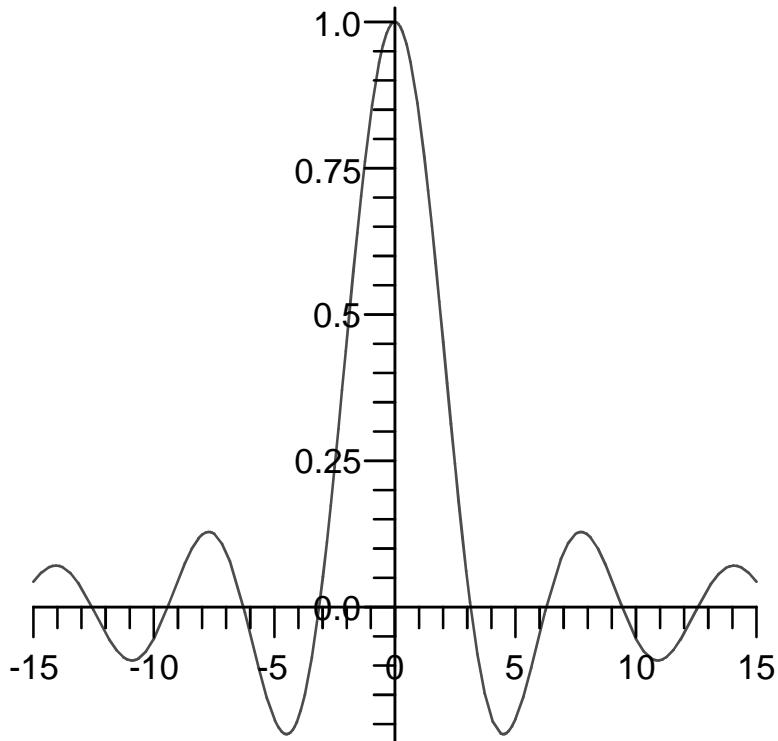
Se llama función seno-cardinal a $f(x) = \frac{\sin(x)}{x}$, $x \in \mathbb{R} - \{0\}$, de ahí su nombre habitual de sinc x.

▼ Apartado a

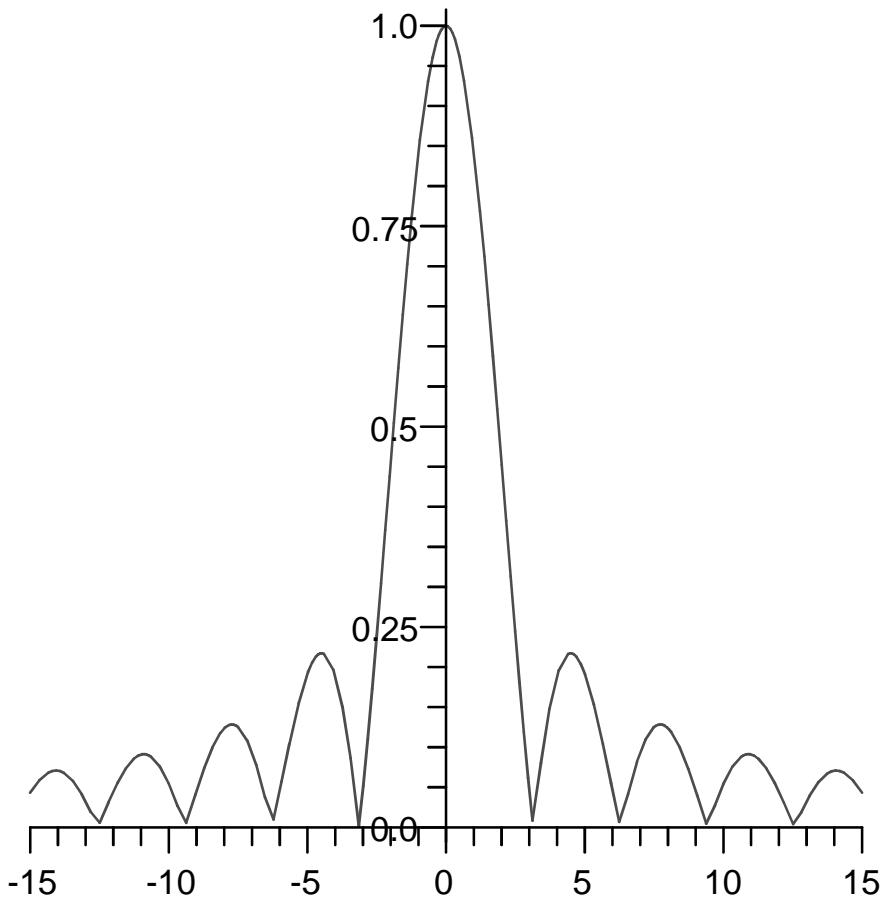
Representar gráficamente $f(x)$ y $|f(x)|$.

▼ Solución

```
> Funcionsinc := x → sin(x)/x :  
> plot(Funcionsinc, -15..15);
```



```
> Valorabsolutosinc := x → abs(sin(x)/x) :  
> plot(Valorabsolutosinc, -15..15);
```



Apartado b

Calcular los máximos y mínimos de la función $f(x)$.

Solución

$$\begin{aligned} &> \text{Derivada} := \text{diff}\left(\frac{\sin(x)}{x}, x\right); \\ &\qquad \text{Derivada} := \frac{\cos(x)}{x} - \frac{\sin(x)}{x^2} \end{aligned} \tag{3.1.1}$$

$$\begin{aligned} &> \text{Derivadasegunda} := \text{diff}(\text{Derivada}, x); \\ &\qquad \text{Derivadasegunda} := -\frac{\sin(x)}{x} - \frac{2 \cos(x)}{x^2} + \frac{2 \sin(x)}{x^3} \end{aligned} \tag{3.1.2}$$

$$\begin{aligned} &> \text{limit}(\text{Derivadasegunda}, x=0); \\ &\qquad \frac{-1}{3} \end{aligned} \tag{3.1.3}$$

$$\begin{aligned} &> \text{solve}(\text{Derivada}=0, x); \\ &\qquad \text{RootOf}(-\tan(_Z) + _Z) \end{aligned} \tag{3.1.4}$$

El máximo absoluto lo tiene en $x=0$, que tiene una discontinuidad evitable, ya que se anula la primera derivada y la segunda es menor que cero en ese punto. Los máximos y mínimos

Locales los tiene en los puntos que satisfacen la ecuación $\tan(x) = x$.

Apartado c

Calcular $\int_0^\infty f(x) dx$

Solución

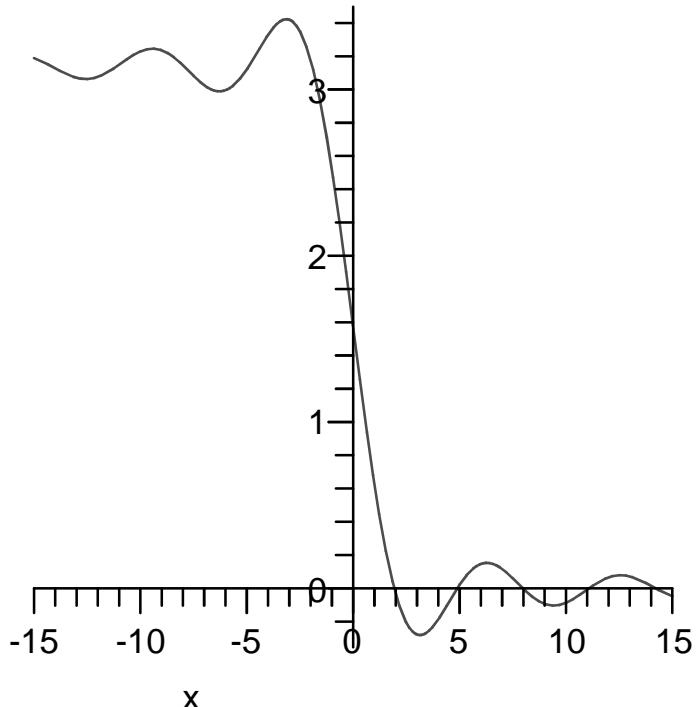
$$\begin{aligned} > \text{int}(\text{Funcionsinc}(x), x=0 .. \infty); \\ &\quad \frac{1}{2} \pi \end{aligned} \tag{4.1.1}$$

Apartado d

Dibujar la función $F(x) = \int_x^\infty f(y) dy$ definida para x real.

Solución

$$\begin{aligned} > \text{Derivadasinc} := x \rightarrow \text{int}(\text{Funcionsinc}(y), y=x.. \infty); \\ &\quad \text{Derivadasinc} := x \rightarrow \int_x^\infty \text{Funcionsinc}(y) dy \\ > \text{plot}(\text{Derivadasinc}(x), x=-15..15); \end{aligned} \tag{5.1.1}$$



▼ Apartado e

Aventurar el valor de $\lim_{y \rightarrow \infty} F(y)$ y de $\lim_{y \rightarrow -\infty} F(y)$ Calcular estos valores.

▼ Solución

$$\begin{cases} > \text{limit}(\text{Derivadasinc}(x), x = \infty); \\ & 0 \end{cases} \quad (6.1.1)$$

$$\begin{cases} > \text{limit}(\text{Derivadasinc}(x), x = -\infty); \\ & \pi \end{cases} \quad (6.1.2)$$

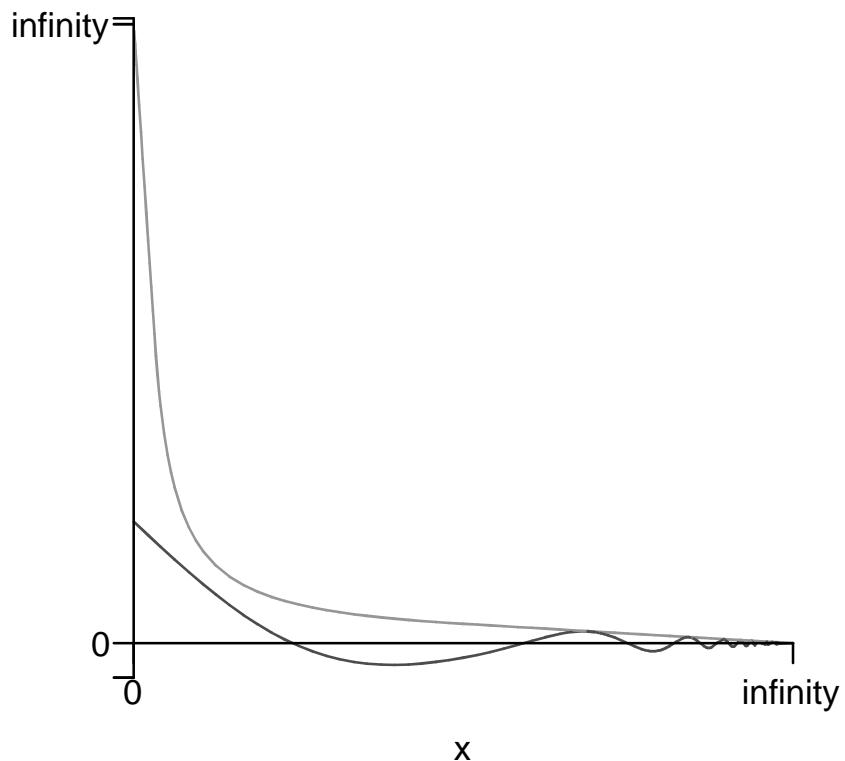
Viendo la gráfica, parece ser que los valores se estabilizan en torno a cero y tres; y exactamente, los valores son cero y pi.

▼ Apartado f

Obtener gráficamente una cota ajustada de $F(y)$ en el intervalo $[0, \infty)$ y comprobar que es una verdadera cota.

▼ Solución

$$> \text{plot}\left(\left[\text{Derivadasinc}(x), \frac{1}{x}\right], x = 0 .. \infty\right);$$



$$\begin{aligned}
 > \text{solve}\left(\text{Derivadasinc}(x) = \frac{1}{x}, x\right); \\
 &\quad \text{RootOf}(2 \text{Si}(_Z) _Z - \pi _Z + 2)
 \end{aligned} \tag{7.1.1}$$

Probando diferentes funciones, vemos que la que más se ajusta gráficamente es la función inversa de x. Vemos que estas dos funciones se cortan en un número determinado de puntos que cumplen la ecuación indicada, siendo Si(x) la función Seno Integral.

▼ Apartado g

Hallar F'(y) en todo punto y real, si existe.

▼ Solución

$$\begin{aligned}
 > \text{DerivadaDerivadasinc} := \text{diff}(\text{Derivadasinc}(x), x); \\
 &\quad \text{DerivadaDerivadasinc} := -\frac{\sin(x)}{x}
 \end{aligned} \tag{8.1.1}$$

$$\begin{aligned}
 > \text{limit}(\text{DerivadaDerivadasinc}, x = 0);
 &\quad -1
 \end{aligned} \tag{8.1.2}$$

Vemos que la función derivada es igual que la primera función porque hemos integrado y luego derivado. Esta función, existe en todo punto real y por tanto F(x) es derivable en todo punto.

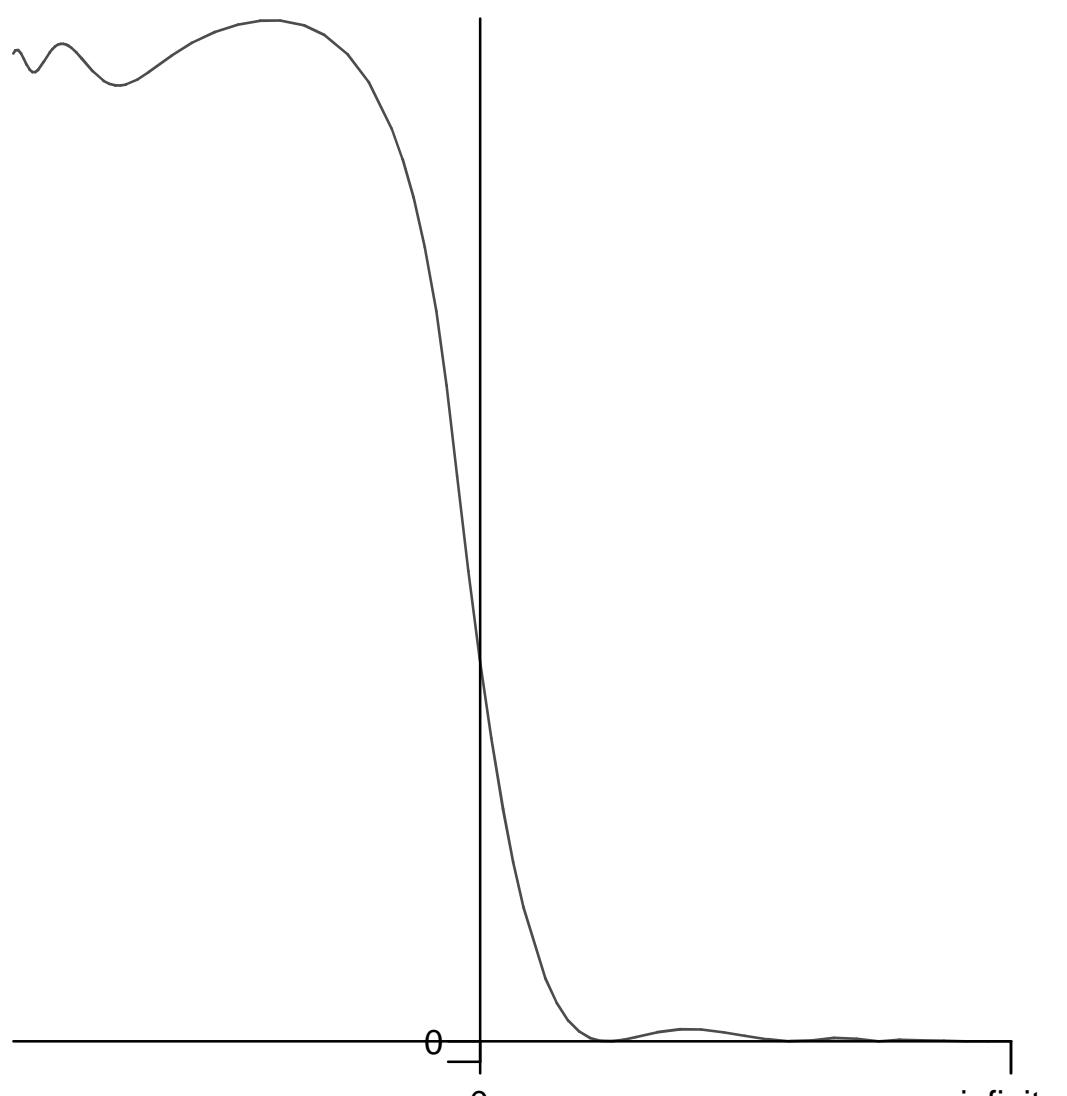
▼ Apartado h

Determinar si F, F' pertenecen a $L_1(R)$

▼ Solución

$$\begin{aligned}
 > \text{Int}\left(\text{abs}(\text{Derivadasinc}(x))^2, x = -infinity.. \infty\right); \\
 &\quad \int_{-\infty}^{\infty} \left| \text{Si}(x) - \frac{1}{2} \pi \right|^2 dx
 \end{aligned} \tag{9.1.1}$$

$$> \text{plot}\left(\text{abs}(\text{Derivadasinc}(x))^2, x = -infinity.. \infty\right);$$



$$= > \int_{-\infty}^0 |Derivadasinc(x)|^2 dx; \quad (9.1.2)$$

Para que una función pertenezca al espacio de Hilbert de orden uno, la integral (o sumatorio) del valor absoluto cuadrático de esa función tiene que ser menor que infinito. Vemos que esta función no cumple esa propiedad, con lo cual, no pertenece al espacio de orden uno de Hilbert.

$$= > \int_{-\infty}^{\infty} |DerivadaDerivadasinc|^2 dx; \quad (9.1.3)$$

Vemos que la función $F'(x)$ tampoco cumple la propiedad de los espacios de Hilbert. Así pues, concluimos que ninguna de las dos pertenecen a dicho espacio.