



Análisis de Fourier

Los orígenes del Análisis de Fourier se remontan a los tiempos de J. Bernoulli y J. Fourier en su intento por resolver los problemas de la cuerda vibrante y de la transmisión de calor, expresando las soluciones como desarrollos en serie de senos y cosenos; expresiones que en aquella época no se consideraban válidas.

En este capítulo se expondrán las definiciones y propiedades más importantes empleadas por la tomografía en su intento por representar las imágenes con la mayor resolución y precisión que le sea posible. Más información puede ser consultada en [3].

A.1 La Transformada de Fourier

No hay unanimidad en la definición de la *Transformada de Fourier*. En esta sección se presentan dos definiciones con sus propiedades asociadas. Ambas formas son muy utilizadas y en este trabajo se usan la dos. Así pues, una primera definición es:

Definición. La *Transformada de Fourier* de una función $f(x) \in L^1(\mathbb{R}^n)$ integrable Lebesgue es

$$\hat{f}(\xi) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} f(x) e^{-i\xi x} dx. \quad (\text{A.1})$$

A. ANÁLISIS DE FOURIER

Además, definimos *La Transformada de Fourier Inversa* de $g(\xi)$ a la función obtenida como

$$\check{g}(x) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} g(\xi) e^{ix\xi} d\xi. \quad (\text{A.2})$$

Así definida, algunas propiedades de las que goza la Transformada de Fourier son:

- *Linealidad* : $(\alpha f + \beta g)^\wedge = \alpha \hat{f} + \beta \hat{g}$.
- $(f * g)^\wedge = (2\pi)^{n/2} \hat{f} \hat{g}$.
- Si $\rho \in O(n)$ es una transformación ortogonal, $(f(\rho x))^\wedge(\xi) = \hat{f}(\rho\xi)$.
- $\int_{\mathbb{R}^n} |f(x)|^2 dx = \int_{\mathbb{R}^n} |\hat{f}(\xi)|^2 d\xi$. Esta ecuación se conoce como *relación de Parseval*.
- $(\frac{\partial^k f}{\partial x^k})^\wedge(\xi) = i^{|k|} \xi^k \hat{f}(\xi)$.
- $(x^k f)^\wedge(\xi) = i^{|k|} \frac{\partial^k \hat{f}}{\partial \xi^k}(\xi)$.
- $\lim_{|\xi| \rightarrow \infty} \hat{f}(\xi) = 0$ (Riemann-Lebesgue). Se deduce de esta propiedad que para funciones integrables las componentes de altas frecuencias tienden a ser despreciables respecto a las de bajas frecuencias.

Otra definición muy usada en la literatura matemática para la *Transformada de Fourier* y su inversa es:

Definición. La *Transformada de Fourier* de una función $f(x) \in L^1(\mathbb{R}^n)$ es

$$\hat{f}(\xi) = \int_{\mathbb{R}^n} f(x) e^{-i2\pi\xi x} dx, \quad (\text{A.3})$$

y su inversa

$$\check{g}(x) = \int_{\mathbb{R}^n} g(\xi) e^{i2\pi x \xi} d\xi. \quad (\text{A.4})$$

Las propiedades anteriores se re-escriben:

- *Linealidad* : $(\alpha f + \beta g)^\wedge = \alpha \hat{f} + \beta \hat{g}$.
- $(f * g)^\wedge = \hat{f} \hat{g}$. La transformada de Fourier del producto de convolución de dos funciones es el producto de sus transformadas.
- Si $\rho \in O(n)$ es una transformación ortogonal, $(f(\rho x))^\wedge(\xi) = \hat{f}(\rho\xi)$.
- $\int_{\mathbb{R}^n} |f(x)|^2 dx = \int_{\mathbb{R}^n} |\hat{f}(\xi)|^2 d\xi$ (Relación de Parseval).
- $(\frac{\partial^k f}{\partial x^k})^\wedge(\xi) = (2\pi)^k i^{|k|} \xi^k \hat{f}(\xi)$.

- $(x^k f)^\wedge(\xi) = (2\pi)^{-k} i^{|k|} \frac{\partial^k \hat{f}}{\partial \xi^k}(\xi)$.
- $\lim_{|\xi| \rightarrow \infty} \hat{f}(\xi) = 0$ (Riemann-Lebesgue).
- Si A es una matriz regular se cumple $\mathfrak{F}(f(Ax)) = \frac{1}{\det(A)} \mathfrak{F}f(A^{-T}\xi)$, siendo \mathfrak{F} la Transformada de Fourier.

Ejemplo. Es muy habitual el uso de la siguiente función:

$$f(x) = \begin{cases} A & \text{si } |x| \leq \tau/2 \\ 0 & |x| > \tau/2 \end{cases}, \quad (\text{A.5})$$

cuya transformada de Fourier según la última definición dada es

$$F(\xi) = A\tau \frac{\text{sen}(\pi\xi\tau)}{\pi\xi\tau}. \quad (\text{A.6})$$

La función $f(x)$, que se llama *pulso rectangular*, es de uso frecuente en el muestreo de señales. Se puede deducir fácilmente que cuanto mayor anchura tiene el pulso más se estrecha su transformada de Fourier; y viceversa. La interpretación física la proporciona la relación de Parseval; la energía de la señal tiende a concentrarse en el lóbulo principal de bajas frecuencias cuando la duración del pulso aumenta. Este resultado es muy importante desde un punto de vista práctico. Si deseamos obtener los detalles en aquellas zonas en las que hay una variación importante de la señal, el pulso deberá estar muy localizado en el espacio o en el tiempo. Los detalles finos del objeto se corresponden con las altas frecuencias y los detalles más groseros con las bajas frecuencias.

A.2 La Transformada de Fourier y la convolución

Se define el *producto de convolución* de dos funciones $f(x)$ y $g(x)$ de \mathbb{R}^n a

$$(f * g)(x) = \int_{\mathbb{R}^n} f(x-y)g(y)dy. \quad (\text{A.7})$$

La operación de convolución se realiza en la práctica para realizar un suavizado en las señales. El fundamento de esta técnica se basa en lo que sucede en el dominio de Fourier. Hacer una convolución en el dominio espacial supone multiplicar las señales en el dominio frecuencial. Si se necesita que la señal $f(x)$ tenga unas propiedades deseadas, como puede ser una frecuencia de corte B , basta multiplicar su transformada de Fourier $F(\xi)$ por una función $H(\xi)$ que se anule para valores $|\xi| \geq B$ y valga la unidad para el resto de valores. A continuación, se realiza la transformación inversa de Fourier para recuperar la señal con las propiedades pretendidas.

A. ANÁLISIS DE FOURIER

Al proceso por el cual una señal se la modifica para que tenga un contenido frecuencial dado se le llama *filtrado de la señal*. Esta es una de las operaciones más importantes en representación de imágenes. Se llama *filtro* a la función con la que se convoluciona la señal.

Para implementar el producto de convolución de dos señales se precisa discretizarlas. Así, definiremos la convolución de dos señales en tiempo o espacio discretas $f(n)$ y $g(n)$ a la señal $h(n)$ en tiempo o espacio discreta

$$h(n) = f(n) * g(n) = \sum_{k=-\infty}^{\infty} f(k)g(n-k). \quad (\text{A.8})$$

Ejemplo. La señal impulso unitario se define como

$$\delta(n) = \begin{cases} 1 & \text{si } n = 0 \\ 0 & \text{si } n \neq 0 \end{cases}. \quad (\text{A.9})$$

Si se convoluciona con cualquier señal en tiempo o espacio discreto se obtiene como resultado la propia señal. Esto se escribe matemáticamente como

$$h(n) = f(n) * \delta(n) = f(n). \quad (\text{A.10})$$

Dado que la señal $h(n) = f(k)\delta(n-k)$ es nula en todo instante salvo $n = k$, cuyo valor es $f(k)$, la ecuación anterior muestra que una señal se puede escribir como combinación infinita de sus muestras.

Definición. Una señal en tiempo o espacio discreta es de *duración finita o longitud N* si se anula en todo instante excepto en N instantes.

Ejemplo. Sea $f(n) = (1, 2)$, entendiendo que el valor de $f(n)$ en el instante n coincide con el valor de la coordenada n empezando en $n = 0$. Con la misma interpretación, sea $g(n) = (1, 2, 3)$. Ambas señales son de longitud finita 2 y 3, respectivamente. Su producto de convolución es

$$h(n) = f(n) * g(n) = (1, 4, 7, 6), \quad (\text{A.11})$$

obtenido como una multiplicación polinómica. En este ejemplo, se obtiene la señal de longitud 4 dada por

$$h(n) = \begin{cases} 1 & \text{si } n = 0 \\ 4 & \text{si } n = 1 \\ 7 & \text{si } n = 2 \\ 6 & \text{si } n = 3 \\ 0 & \text{en otro caso.} \end{cases} \quad (\text{A.12})$$

En general, dadas dos señales de longitudes N y M su producto de convolución tendrá longitud $N + M - 1$.

A.3 Muestreo de señales

Al muestrear una señal en tiempo continuo $f(x)$ se obtiene una señal en tiempo discreto f_n obtenida a partir de la primera mediante la fórmula $f_n = f(\tau n)$, siendo $1/\tau$ la *velocidad de muestreo*. Dicho de otro modo, escogemos valores de $f(x)$ a intervalos de τ unidades.

El objetivo de la teoría del muestreo es la recuperación de la señal original a partir de sus muestras. Se plantea las condiciones que deben poseer las señales $f(x)$ para que esto sea posible. Además, propone fórmulas que puedan ser implementadas en la práctica con los medios de cálculo de que se dispone.

Se dice que $f(x)$ tiene ancho de banda B si su transformada de Fourier es localmente integrable y es nula para frecuencias $|\xi| \geq B$. Luego, la frecuencia más grande - llamada *frecuencia de corte* - que tiene $f(x)$ es B .

De entre las señales de ancho de banda limitada sobresale en importancia la función

$$\text{sinc}(x) = \begin{cases} \frac{\text{sen}(x)}{x} & \text{si } x \neq 0 \\ 1 & \text{si } x = 0, \end{cases} \quad (\text{A.13})$$

que se obtiene mediante la transformada inversa de Fourier de

$$F(\xi) = \begin{cases} 1 & \text{si } |\xi| \leq 1 \\ 0 & \text{otro caso.} \end{cases} \quad (\text{A.14})$$

Por lo tanto, la señal $f(x)$ tiene ancho de banda 1. A partir de ella, se define en \mathbb{R}^n la función

$$\text{sinc}(x) = \text{sinc}(x_1) \dots \text{sinc}(x_n), \quad (\text{A.15})$$

con $x = (x_1, \dots, x_n)^T$. Y finalmente, $\text{sinc}_b(x) = \text{sinc}(bx)$, cuya transformada de Fourier es

$$\text{sinc}_b(\xi) = (\pi/2)^{n/2} b^{-n} \aleph(\frac{1}{b}\xi), \quad (\text{A.16})$$

donde \aleph es la función característica de $[-1, 1]^n$.

Esta función juega un papel importante en la recuperación de la señal a partir de sus muestras. Supongamos que $f(x)$ es una función que se anula fuera de un intervalo $[-a, a]$. Luego, $f(x) = 0$ para $|\xi| > a$. Si extendemos por periodicidad $f(x)$ a todo \mathbb{R} se podrá escribir como una combinación de exponenciales complejas cuyos coeficientes son los coeficientes de Fourier; y éstos se relacionan con la transformada de Fourier.

$$f(x) = \sum_k f_k e^{i\pi x k/a}, \quad (\text{A.17})$$

siendo los coeficientes de Fourier

$$f_k = (2a)^{-n} \int_{[-a, a]} f(x) e^{-i\pi x k/a} dx, \quad (\text{A.18})$$

A. ANÁLISIS DE FOURIER

y claramente

$$f_k = \left(\frac{\pi}{2}\right)^{n/2} a^{-n} \sum_k \hat{f}\left(\frac{\pi}{a}k\right). \quad (\text{A.19})$$

Si se aplica este resultado a una señal de cuadrado integrable, con ancho de banda B y para $h \leq \frac{\pi}{B}$, se deduce que su transformada de Fourier es

$$\hat{f}(\xi) = (2\pi)^{-n/2} h^n \sum_k f(hk) e^{-ih\xi k}. \quad (\text{A.20})$$

Para recuperar $f(x)$ basta multiplicar por la función característica $\mathbb{N}_{1/B}$ de $[-(\pi/h), \pi/h]$ y hacer la transformada inversa de Fourier. Se llega a la fórmula

$$f(x) = \sum_k f(hk) \operatorname{sinc} \frac{\pi}{h}(x - hk). \quad (\text{A.21})$$

Nota. La condición $h \leq \frac{\pi}{B}$ se llama *condición de Nyquist*.

Ejemplo. Sea $f(x) = 2 \sin(4\pi x) + 5 \sin(10\pi x)$ para $0 \leq x \leq 1$. Se observa que la máxima frecuencia de la señal es de 5 Hz. Se ha representado la gráfica de la función y las gráficas que se obtienen al interpolar con tasas de muestreo inferior, igual y superior a la de Nyquist. Claramente, tasas de velocidad de muestreo inferiores a la de Nyquist producen resultados insatisfactorios. Es importante darse cuenta de que el muestreo a la tasa de Nyquist no produce resultados correctos en este caso. La señal original se recupera a una velocidad de muestreo superior a la de Nyquist, como bien se aprecia en las gráficas de la figura A.1.

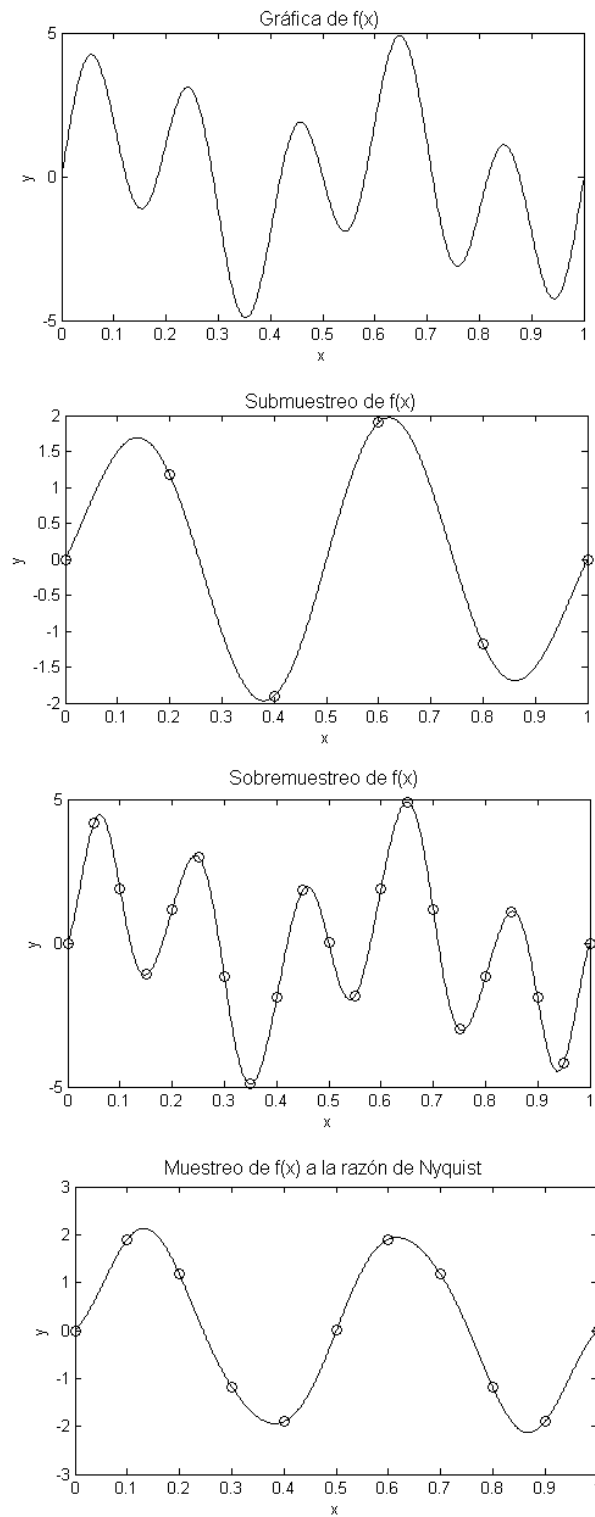


Figura A.1: Ejemplo de muestreo de una señal

B

Inversión de la Transformada de Radon

El objetivo es recuperar el objeto $f(x)$ a partir de sus integrales de línea. Para ello, una herramienta útil va a ser el potencial de Riesz y la transformada dual de Radon [9].

Definición. Para $\alpha < n$, se define el operador lineal I^α que cumple

$$(I^\alpha f)^\wedge(\xi) = |\xi|^{-\alpha} \hat{f}(\xi). \quad (\text{B.1})$$

Si $f(x)$ pertenece a la clase de funciones de Schwartz entonces se garantiza la existencia de inversa para el operador lineal, que es $I^{-\alpha}$. Por lo tanto, $I^{-\alpha} I^\alpha f = f$.

Ahora, de la propia definición del potencial de Riesz,

$$I^\alpha f(x) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} |\xi|^{-\alpha} \hat{f}(\xi) e^{-ix \cdot \xi} d\xi. \quad (\text{B.2})$$

Y realizando el cambio a coordenadas polares $\xi = \rho \cdot w$, se obtiene

$$I^\alpha f(x) = (2\pi)^{-n/2} \int_{S^{n-1}} \int_0^\infty \rho^{n-1-\alpha} \hat{f}(\rho \cdot w) e^{-i\rho x \cdot w} d\rho dw. \quad (\text{B.3})$$

Aplicando el Teorema de la Rebanada de Fourier y que la Transformada de Radon posee simetría par se llega a la siguiente expresión

$$I^\alpha f(x) = 1/2(2\pi)^{-n+1/2} \int_{S^{n-1}} \int_{-\infty}^\infty |\rho|^{n-1-\alpha} (Rf)^\wedge(w, \rho) e^{-i\rho x \cdot w} d\rho dw. \quad (\text{B.4})$$

B. INVERSIÓN DE LA TRANSFORMADA DE RADON

En la expresión anterior aparece el potencial de Riesz en la integral. Puede escribirse ahora

$$I^\alpha f(x) = 1/2(2\pi)^{-n+1} \int_{S^{n-1}} I^{\alpha+1-n} Rf(w, x \cdot w) dw. \quad (\text{B.5})$$

La integral es la transformada dual de Radon. Se aplica $I^{-\alpha}$ y $f(x)$ queda despejada como

$$f = 1/2(2\pi)^{1-n} I^{-\alpha} R^\# I^{\alpha-n-1} Rf. \quad (\text{B.6})$$

A partir de aquí pueden obtenerse fórmulas interesantes particularizando para diferentes valores de α .

Caso 1. Si $\alpha = 0$ se obtiene la fórmula explícita

$$f = \frac{1}{2}(2\pi)^{1-n} R^\# I^{1-n} Rf. \quad (\text{B.7})$$

Para g perteneciente a la clase de Schwartz se tiene

$$(I^{1-n} g)^\wedge(\xi) = [\text{sgn}(\xi)]^{n-1} \xi^{n-1} \hat{g}(\xi). \quad (\text{B.8})$$

Teniendo en cuenta la definición de la *transformada de Hilbert* de una función f de la clase de Schwartz como aquella que cumple

$$(Hf)^\wedge(\xi) = -i \text{sgn} \xi \hat{f}(\xi), \quad (\text{B.9})$$

o equivalentemente

$$Hf = \frac{1}{\pi} v.p. \frac{1}{x} * f, \quad (\text{B.10})$$

y que verifica $H(Hf) = -f$, se tiene que $I^{1-n} g = H^{n-1} g^{(n-1)}$. Por lo tanto, se puede escribir con un poco de cálculo

$$f(x) = \frac{1}{2}(2\pi)^{1-n} \cdot \begin{cases} (-1)^{\frac{n-2}{2}} \int_{S^{n-1}} H(Rf)^{(n-1)}(w, x \cdot w) dw & \text{si } n \text{ es par} \\ (-1)^{\frac{n-1}{2}} \int_{S^{n-1}} (Rf)^{(n-1)}(w, x \cdot w) dw & \text{si } n \text{ es impar} \end{cases}. \quad (\text{B.11})$$

Para el caso n par y escribiendo la transformada de Hilbert como convolución se obtiene, denotando $g = Rf$, que f es

$$f(x) = (-1)^{n/2+1} (2\pi)^{-n} \int_{S^{n-1}} v.p. \int_{\mathbb{R}} \frac{g^{(n-1)}(w, s)}{x \cdot w - s} ds dw. \quad (\text{B.12})$$

Realizando el cambio de variable $s = q + x \cdot w$ y que para el valor principal de Cauchy es cierto que

$$v.p. \int_{\mathbb{R}} \frac{f(x)}{x} dx = \lim_{h \rightarrow 0} \int_{|x| > h} \frac{f(x)}{x} dx = \int_{-\infty}^{\infty} \frac{f(x) - f(-x)}{2x} dx, \quad (\text{B.13})$$

se escribe

$$f(x) = (-1)^{n/2} (2\pi)^{-n} \frac{1}{2} \int_{\mathbb{R}} \frac{1}{q} \int_{S^{n-1}} (g^{(n-1)}(w, x \cdot w + q) - g^{(n-1)}(w, x \cdot w - q)) dw dq. \quad (\text{B.14})$$

Puesto que g tiene simetría par y $n - 1$ es impar entonces $g^{(n-1)}$ es impar, y se puede simplificar la expresión anterior. Así, definiendo

$$F_x(q) = \frac{1}{|S^{n-1}|} \int_{S^{n-1}} g(w, x \cdot w + q) dw, \quad (\text{B.15})$$

que es par por serlo g , se llega a la fórmula original de Radon del año 1917

$$f(x) = 2(-1)^{n/2} (2\pi)^{-n} |S^{n-1}| \int_0^\infty \frac{1}{q} F_x^{(n-1)}(q) dq. \quad (\text{B.16})$$

Particularizando para el caso plano ($n = 2$)

$$f(x) = -\frac{1}{\pi} \int_0^\infty \frac{dF_x(q)}{q} dq. \quad (\text{B.17})$$

Caso 2. Si $\alpha = n - 1$ se obtiene para $g = Rf$

$$f = \frac{1}{2} (2\pi)^{1-n} I^{1-n} R^\sharp g. \quad (\text{B.18})$$

Si n es impar el potencial de Riesz verifica $I^{1-n} = (-\Delta)^{(n-1)/2}$ y particularizando al caso $n = 3$ se llega a la fórmula explícita

$$f(x) = \frac{-1}{8\pi^2} \Delta \int_{S^2} g(w, x \cdot w) dw, \quad (\text{B.19})$$

donde el operador laplaciano actúa en la variable x .

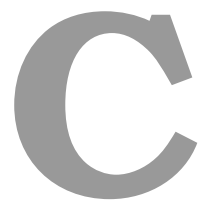
Otra forma de conseguir una fórmula de inversión, y que será adecuada para obtener posteriormente un algoritmo para recuperar el objeto, parte de la fórmula ya vista

$$(R^\sharp w) * f = R^\sharp (w * Rf). \quad (\text{B.20})$$

Si se elige una función g cuya transformada dual de Radon $R^\sharp w$ se aproxime a la distribución δ de Dirac se obtendrá f como límite. Como se supone que las señales tienen frecuencia de corte B podemos escribir

$$(R^\sharp w)^\wedge(\xi) = (2\pi)^{-n/2} \hat{\Phi}(|\xi|/B), \quad (\text{B.21})$$

con $0 \leq \hat{\Phi} \leq 1$ y $\hat{\Phi}(\rho) = 0$ para $\rho \geq 1$. La función $\hat{\Phi}(\rho)$ se puede interpretar como un filtro paso bajo para la señal.



Tomografía matemática. Difracción

Se considera el caso en el que la trayectoria de los rayos no es rectilínea y sufre del fenómeno de la difracción. Es lo que ocurre cuando se usan otro tipo de fuentes diferentes a la de los rayos-X y que son más seguros, sobre todo si su exposición a ellos es a bajas dosis; ondas electromagnéticas y ultrasonidos entre ellos. El parámetro a reconstruir y que se considera como el objeto es el índice de refracción del medio; ya sea el electromagnético o el acústico.

Se puede tratar el problema de dos maneras diferentes:

1. Se estima un valor inicial para el índice de refracción y se averigua el camino o trayectoria que debería seguir cada rayo dentro del espécimen. Se compara con los resultados de laboratorio obtenidos y se efectúa la corrección correspondiente en el índice de refracción. El proceso se vuelve a repetir calculando las nuevas trayectorias con el valor corregido. Se corrige el índice y así hasta que se llegue a cumplir el criterio de parada establecido. En estos métodos se utilizan técnicas algebraicas y no se comporta mal cuando los casos de estudio son débilmente difractantes.
2. La teoría geométrica de rayos es adecuada para heterogeneidades cuyos tamaños son mayores que la longitud de onda del medio con el que se desea ver. En otro caso, se debe tratar el problema acudiendo a la ecuación de ondas y teniendo en cuenta el fenómeno de difracción.

C. TOMOGRAFÍA MATEMÁTICA. DIFRACCIÓN

Si estos fenómenos de difracción son débiles, se puede hacer una reconstrucción tomográfica basada en el *Teorema de Difracción de Fourier*.

Para hacer tratable el problema se estudia en primer lugar el caso de materiales homogéneos para pasar, a continuación, a estudiar el caso de muestras no homogéneas. El material consultado en este apéndice y que se anima al lector a que profundice en su lectura es [5].

C.1 Caso homogéneo

Sea u la magnitud del campo expresado en función de la posición \vec{r} y del tiempo t . Sea c la velocidad del campo. La ecuación de ondas se escribe

$$\nabla^2 u(\vec{r}, t) - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} u(\vec{r}, t) = 0. \quad (\text{C.1})$$

Supongamos que la onda

$$u(\vec{r}, t) = u(\vec{r}) e^{-j\omega t} \quad (\text{C.2})$$

verifica la ecuación C.1. Definiendo el número de ondas como

$$k(\vec{r}) = \frac{2\pi}{\lambda} = \frac{\omega}{c}, \quad (\text{C.3})$$

se obtiene

$$[\nabla^2 + k^2(\vec{r})] u(\vec{r}) = 0. \quad (\text{C.4})$$

Para un medio que es homogéneo el número de ondas es constante, $k(\vec{r}) = k_0$, y así

$$(\nabla^2 + k_0^2) u(\vec{r}) = 0. \quad (\text{C.5})$$

Como solución de prueba ensayamos una onda plana

$$u(\vec{r}) = e^{j\vec{k} \cdot \vec{r}} = e^{j(k_x x + k_y y)}. \quad (\text{C.6})$$

Nota. Se denomina *vector de propagación* a $k(\vec{r}) = (k_x, k_y)$.

Ahora, deberá cumplirse

$$\frac{\partial^2 u(\vec{r})}{\partial x^2} + \frac{\partial^2 u(\vec{r})}{\partial y^2} + k_0^2 u(\vec{r}) = 0, \quad (\text{C.7})$$

con

$$|k(\vec{r})|^2 = k_x^2 + k_y^2 = k_0^2. \quad (\text{C.8})$$

La solución general de la ecuación de ondas homogénea, debido a la linealidad, será una suma ponderada de todas las soluciones de ondas planas posibles. Luego, para una frecuencia temporal ω , el campo $u(\vec{r})$ puede escribirse

$$u(\vec{r}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \alpha(k_y) e^{j(k_x x + k_y y)} dk_y + \frac{1}{2\pi} \int_{-\infty}^{\infty} \beta(k_y) e^{j(-k_x x + k_y y)} dk_y, \quad (C.9)$$

con

$$k_x = \sqrt{k_0^2 - k_y^2}. \quad (C.10)$$

En la expresión anterior $\alpha(k_y)$ y $\beta(k_y)$ son los coeficientes de las ondas que viajan hacia la derecha y hacia la izquierda, respectivamente. Considerando el caso en el que el emisor está a la izquierda y el detector se coloca a la derecha sólo se ha de retener el primer sumando. Así, la solución es

$$u(\vec{r}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \alpha(k_y) e^{j(k_x x + k_y y)} dk_y. \quad (C.11)$$

Si se coloca el detector en $x = 0$ obtendremos la expresión

$$u(0, y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \alpha(k_y) e^{jk_y y} dk_y. \quad (C.12)$$

Por lo tanto, se llega a que el coeficiente es la transformada inversa de Fourier de la solución en la línea de medición $x = 0$

$$\alpha(k_y) = \mathcal{F}^{-1}(u(0, y)). \quad (C.13)$$

Conocido el coeficiente, éste se introduce en la ecuación C.11 y se halla el valor en cualquier lugar a la derecha de $x = 0$. Ahora es fácil obtener una expresión que ligue los valores del campo cuando la línea de detectores se coloca en dos posiciones paralelas diferentes desplazadas entre ellas una distancia $d = l_1 - l_0$. Se obtiene

$$u(l_1, y) = u(l_0, y) e^{jk_x d}. \quad (C.14)$$

Se observa que lo que sucede es un desplazamiento de fase.

Se puede sistematizar el proceso de la siguiente manera:

1. Hacer la transformada de Fourier de $u(l_0, y)$ para obtener la descomposición de u como función de k_y .
2. Propagar cada onda plana a la línea $x = l_1$ multiplicando su amplitud compleja por el factor $e^{jk_x d}$, siendo $d = l_1 - l_0$ y $k_x = \sqrt{k_0^2 - k_y^2}$.
3. Hacer la transformada de Fourier inversa de la descomposición de ondas planas y obtener el valor del campo en $u(l_1, y)$.

C.2 Caso no-homogéneo

Para obtener la solución en el caso de medios heterogéneos se parte de la ecuación de ondas para pequeñas perturbaciones

$$(\nabla^2 + k_0^2) u(\vec{r}) = -o(\vec{r})u(\vec{r}), \quad (\text{C.15})$$

siendo el objeto $o(\vec{r})$ el que está relacionado con las propiedades de la muestra y cuyo valor es

$$o(\vec{r}) = k_0^2 [n^2(\vec{r}) - 1], \quad (\text{C.16})$$

siendo $n(\vec{r})$ el índice de refracción correspondiente. Escribimos el campo como suma de una componente solución del caso homogéneo $u_0(\vec{r})$ y otra la debida a las heterogeneidades o campo dispersado $u_s(\vec{r})$. Tenemos

$$u(\vec{r}) = u_0(\vec{r}) + u_s(\vec{r}). \quad (\text{C.17})$$

Puesto que $u(\vec{r})$ es solución, se debe cumplir la *ecuación de Helmholtz*

$$(\nabla^2 + k_0^2) u_s(\vec{r}) = -o(\vec{r})u(\vec{r}), \quad (\text{C.18})$$

cuya solución se puede expresar en términos de la función de Green y que es solución, a su vez, de la ecuación diferencial

$$(\nabla^2 + k_0^2) g(\vec{r}, \vec{r}') = -\delta(\vec{r} - \vec{r}'). \quad (\text{C.19})$$

Se sabe que la solución de la ecuación C.19 es:

- para el caso 2-dimensional

$$g(\vec{r}, \vec{r}') = \frac{j}{4} H_0^{(1)}(k_0 R), \quad (\text{C.20})$$

con $R = |\vec{r} - \vec{r}'|$ y $H_0^{(1)}$ la función de *Hankel* de orden cero y de primera clase;

- para el caso 3-dimensional,

$$g(\vec{r}, \vec{r}') = \frac{e^{jk_0 R}}{4\pi R}. \quad (\text{C.21})$$

Se observa que en ambos casos $g(\vec{r}, \vec{r}')$ es función de $\vec{r} - \vec{r}'$ y así, g también lo es. Sólo queda expresar como suma de impulsos

$$o(\vec{r})u(\vec{r}) = \int o(\vec{r}')u(\vec{r}')\delta(\vec{r} - \vec{r}')d\vec{r}', \quad (\text{C.22})$$

y escribir la ecuación integral (integral de Fredholm de segunda clase)

$$u_s(\vec{r}) = \int g(\vec{r} - \vec{r}')o(\vec{r}')u(\vec{r}')d\vec{r}'. \quad (\text{C.23})$$

C.3 Aproximación de Born

Suponiendo la condición $u_s(\vec{r}) \ll u_0(\vec{r})$ y escribiendo el segundo miembro de la ecuación C.23 como suma de dos integrales

$$u_s(\vec{r}) = \int g(\vec{r} - \vec{r}') o(\vec{r}') u_0(\vec{r}') d\vec{r}' + \int g(\vec{r} - \vec{r}') o(\vec{r}') u_s(\vec{r}') d\vec{r}', \quad (\text{C.24})$$

se desprecia la contribución de la segunda integral y queda

$$u_s(\vec{r}) \approx u_B(\vec{r}) = \int g(\vec{r} - \vec{r}') o(\vec{r}') u_0(\vec{r}') d\vec{r}'. \quad (\text{C.25})$$

El proceso se puede iterar sustituyendo en C.25 la aproximación u_B obtenida dentro de la integral y hallar $u_B^{(2)}(\vec{r})$. Así,

$$u_B^{(2)}(\vec{r}) = \int g(\vec{r} - \vec{r}') o(\vec{r}') [u_0(\vec{r}') + u_B(\vec{r}')] d\vec{r}'. \quad (\text{C.26})$$

Y así, sucesivamente, podemos escribir

$$u_B^{(i+1)}(\vec{r}) = \int g(\vec{r} - \vec{r}') o(\vec{r}') [u_0(\vec{r}') + u_B^{(i)}(\vec{r}')] d\vec{r}'. \quad (\text{C.27})$$

C.4 Teorema de Difracción de Fourier

La ecuación C.25 se puede interpretar como una convolución entre la función de Green $g(\vec{r} - \vec{r}')$ y el producto $o(\vec{r}') u_0(\vec{r}')$. Para el caso que la onda incidente sea una onda plana, siendo $\vec{k} = (k_x, k_y)$,

$$u_0(\vec{r}) = e^{j\vec{k} \cdot \vec{r}}, \quad (\text{C.28})$$

satisfaciendo $k_0^2 = k_x^2 + k_y^2$. Y puesto que la solución para $g(\vec{r}, \vec{r}')$ viene dada por C.20 puede escribirse explícitamente

$$g(\vec{r}, \vec{r}') = \frac{j}{4\pi} \int_{-\infty}^{\infty} \frac{1}{\beta} e^{j[\alpha(x-x') + \beta|y-y'|]} d\alpha, \quad (\text{C.29})$$

siendo $\beta = \sqrt{k_0^2 - \alpha^2}$. Sustituyendo la expresión C.29 en la ecuación C.25 se deduce

$$u_B(\vec{r}) = \frac{j}{4\pi} \int o(\vec{r}') u_0(\vec{r}') \int_{-\infty}^{\infty} \frac{1}{\beta} e^{j[\alpha(x-x') + \beta|y-y'|]} d\alpha d\vec{r}'. \quad (\text{C.30})$$

Por conveniencia, supongamos que la dirección de propagación de la onda plana incidente es el eje y positivo. Luego, $u_0(\vec{r}) = e^{j\vec{s}_0 \cdot \vec{r}}$, con $\vec{s}_0 = (0, k_0)$. Y supongamos, además, que la línea de

C. TOMOGRAFÍA MATEMÁTICA. DIFRACCIÓN

receptores está en la posición $y = l_0$ y, por lo tanto, $l_0 - y > 0$ para todo punto interior de la muestra. Entonces podemos escribir

$$u_B(x, y = l_0) = \frac{j}{4\pi} \int_{-\infty}^{\infty} d\alpha \int \frac{o(\vec{r}')}{\beta} e^{j[\alpha(x-x')+\beta(y-y')]} e^{jk_0 y'} d\vec{r}'. \quad (\text{C.31})$$

Denotamos con $O(\alpha, \beta - k_0)$ la transformada de Fourier del objeto $o(\vec{r})$ a frecuencias α y $\beta - k_0$ y escribimos

$$u_B(x, y = l_0) = \frac{j}{4\pi} \int_{-\infty}^{\infty} \frac{1}{\beta} e^{j(\alpha x + \beta l_0)} O(\alpha, \beta - k_0) d\alpha. \quad (\text{C.32})$$

Denotamos por $U_B(\omega, l_0)$ la transformada de Fourier de u_B ;

$$U_B(\omega, l_0) = \int_{-\infty}^{\infty} u_B(x, l_0) e^{-j\omega x} dx. \quad (\text{C.33})$$

Si se tiene en cuenta la expresión dada para U_B , que se verifica

$$\int_{-\infty}^{\infty} e^{j(\omega - \alpha)x} dx = 2\pi\delta(\omega - \alpha) \quad (\text{C.34})$$

y teniendo en cuenta la ecuación C.32, se llega finalmente a la expresión

$$U_B(\alpha, l_0) = \frac{j}{2\sqrt{k_0^2 - \alpha^2}} e^{j\sqrt{k_0^2 - \alpha^2} l_0} O(\alpha, \sqrt{k_0^2 - \alpha^2} - k_0); \quad (\text{C.35})$$

válida cuando $|\alpha| < k_0$. Conviene decir que la física de propagación de ondas da escasa probabilidad a que la mayor frecuencia angular del campo dispersado medido en la línea $y = l_0$ exceda el valor de k_0 . De manera que en la mayoría de las situaciones prácticas $U_s(\omega, l_0)$ será nula para valores $\omega > k_0$.

El factor $\frac{j}{2\sqrt{k_0^2 - \alpha^2}} e^{j\sqrt{k_0^2 - \alpha^2} l_0}$ es una constante para una línea fija de receptores. Cuando α varía de $-k_0$ a k_0 las coordenadas $(\alpha, \sqrt{k_0^2 - \alpha^2} - k_0)$ de la transformada de Fourier de la función objeto trazan un arco semicircular en el plano (u, v) . Los puntos extremos del arco semicircular están a una distancia $\sqrt{2}k_0$ del origen en el dominio de la frecuencia. Para $|\alpha| > k_0$ la transformada es nula.

Para un objeto cuyas heterogeneidades son débilmente dispersantes podemos establecer el siguiente resultado.

Teorema de Difracción de Fourier. Si un objeto definido por la función $o(x, y)$ se ilumina con una onda plana entonces la transformada de Fourier del campo dispersado por el objeto, medido sobre una línea en el sentido de avance, coincide con los valores de la transformada de Fourier 2-dimensional del objeto, $O(\omega_1, \omega_2)$, a lo largo de un arco semicircular en el dominio de la frecuencia (figura C.1).

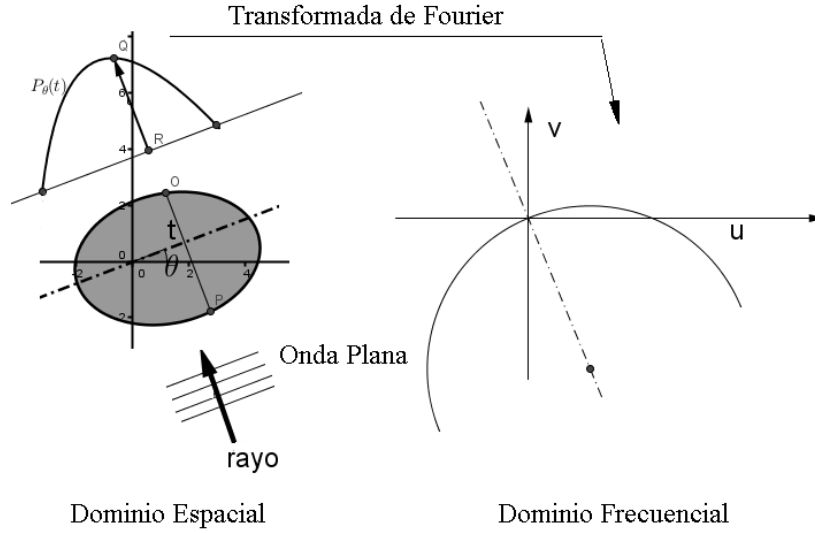


Figura C.1: Relación entre la transformada de Fourier de la proyección y la transformada de Fourier del objeto.

El radio de los arcos viene dado por $k_0 = \frac{2\pi}{\lambda}$, siendo λ la longitud de onda. Se observa que al disminuir la longitud de onda aumenta el radio de los arcos y, por lo tanto, el Teorema de Difracción de Fourier se aproxima al Teorema de la Rebanada de Fourier.

Según el teorema de Difracción de Fourier, para que la reconstrucción de un objeto fuera perfecta debería conocerse el valor de la transformada de Fourier en todos los puntos interiores del disco centrado en el origen y de radio $\sqrt{2}K_0$. Eligiendo de forma adecuada la orientación y la frecuencia de las ondas planas incidentes es posible estimar la transformada de Fourier del objeto a cualquier frecuencia. Por ejemplo, variando la frecuencia del campo incidente conseguimos cambiar el valor del radio de los arcos. El objetivo es conseguir generar una estimación de la transformada de Fourier completa del objeto.

Rotando el objeto y midiendo el campo dispersado para diferentes orientaciones es posible obtener esas estimaciones. Con la rotación del objeto se rotan los arcos en el dominio de Fourier.

Un problema importante es la interpolación que debe realizarse. Nótese que los datos en el dominio de Fourier se disponen sobre arcos semicirculares y se deben aproximar a una red rectangular para realizar la inversión que proporcione los valores del objeto (figura C.2).

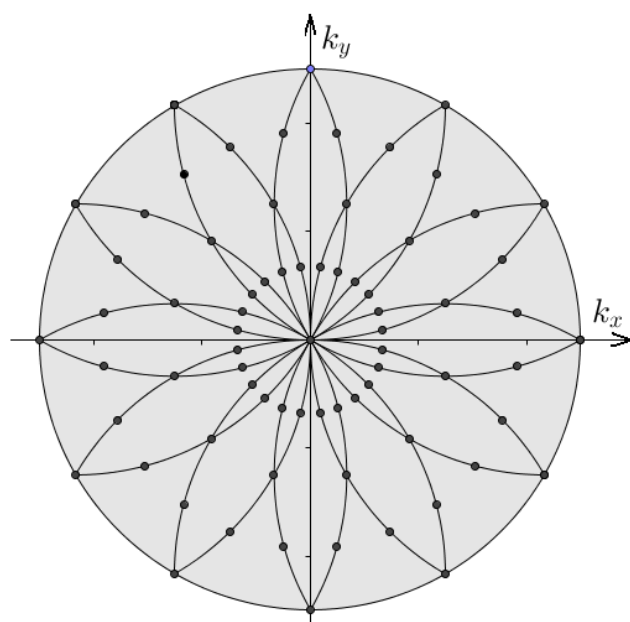


Figura C.2: Los datos medidos se disponen sobre los arcos semicirculares.

D

Programas

Los algoritmos realizados por el autor de este proyecto se recopilan en este apéndice. Todas las figuras del capítulo correspondiente a la transformada de Radon han sido realizados con los mismos. Se han programado con MATLAB 7.6 todos los algoritmos relacionados con la transformada de Radon. El último de los programas es una implementación en Turbo Pascal de la FFT en el que se ha usado el algoritmo de Buneman para la re-ordenación de los nodos.

D.1 Programas

1. Algoritmo que define el fantasma de 10 elipses para realizar la simulaciones.

```
% Descripción de las elipses del phantom
% elipses=phantomas(total);
% total es el número de elipses a dibujar
function [elipses]=phantomas(total)
ellipse(1,:)= [1 0.69 0.92 0 0 0];
ellipse(2,:)= [-0.8 0.6624 0.8740 0 -0.0184 0];
ellipse(3,:)= [-0.2 0.11 0.31 0.22 0 -18];
ellipse(4,:)= [-0.2 0.16 0.41 -0.22 0 18];
ellipse(5,:)= [0.1 0.21 0.25 0 0.35 0];
ellipse(6,:)= [0.1 0.046 0.046 0 0.1 0];
ellipse(7,:)= [0.1 0.046 0.046 0 -0.1 0];
ellipse(8,:)= [0.1 0.046 0.023 -0.08 -0.605 0];
ellipse(9,:)= [0.1 0.023 0.023 0 -0.606 0];
ellipse(10,:)= [0.1 0.023 0.046 0.06 -0.605 0];
elipses=[];
for numero=1:total
    elipses=[elipses;ellipse(numero,:)];
end
```

2. Algoritmo que calcula la proyección de una elipse cualquiera

```
function [P] = proyeccion(ellipse,theta,coord)
% Proyeccion de una elipse de centro cualquiera
% y ejes cualesquiera
% ellipse corresponde a los datos de la elipse a
% proyectar
% theta es el ángulo de proyección
% coord es la coordenada sobre la recta que
% recorren los rayos

% datos de la elipse
rho=ellipse(1);
A=ellipse(2);
B=ellipse(3);
x1=ellipse(4);
y1=ellipse(5);
alfa=ellipse(6);
```

```
% cálculo de gamma y s
gamma=(180/pi)*angle(x1+i*y1);
s=sqrt(x1^2+y1^2);

% cálculo del ángulo y de la coordenada
% modificada
theta_modif=theta-alfa;
t_modif=coord-s*cos((pi/180)*(gamma-theta));

% cálculo de a
a2=A^2*(cos((pi/180)*theta_modif))^2+
B^2*(sin((pi/180)*theta_modif))^2;
a=sqrt(a2);

% cálculo de P(t)
if (abs(t_modif)>a)
    P=0;
else
    P=(2*rho*A*B/a2)*sqrt(a2-t_modif^2);
end
end
```

3. Algoritmo que calcula la proyección de un conjunto de elipses.

```
function [y]=proyeccion_objeto(objeto,
angulo,coord)
y=0;
tamano=size(objeto);
total=tamano(1);
for numero_elipse=1:total
    elipse=objeto(numero_elipse,:);
    y=y+proyeccion(elipse,angulo,coord);
end
end
```

4. Algoritmo que calcula la gráfica de la proyección de una elipse.

```
function [x,y]=grafica_proyeccion_elipse(elipse,
angulo,puntos)
incremento=2/(puntos-1);
y=[];
```

D. PROGRAMAS

```
x=[];  
for t=-1:incremento:1  
    f=proyeccion(ellipse, angulo, t);  
    y=[y, f];  
    x=[x, t];  
end  
end
```

5. Algoritmo que calcula la gráfica de la proyección de un objeto.

```
function [x,y]=grafica_proyeccion_objeto(objeto,  
angulo,q)  
h=1/q;  
y=[];  
x=[];  
for l=-q:1:q  
    sl=l*h;  
    [f]=proyeccion_objeto(objeto, angulo, sl);  
    y=[y, f];  
    x=[x, sl];  
end  
end
```

6. Algoritmo que calcula el filtro de RamLak

```
function [w]=RamLak(h,l)  
% Filtro de Ram y Lak  
if l==0  
    w=1/(8*h^2);  
elseif (mod(l,2)~=0)  
    w=-1/(2*pi^2*h^2*l^2);  
else  
    w=0;  
end  
end
```

7. Algoritmo que calcula el filtro de Shepp-Logan.

```
function [w]=SheppLogan(c,s);  
%filtro Shepp Logan  
numerador=pi/2-c*s*sin(c*s);  
denominador=(pi/2)^2-(c*s)^2;
```



```
    if (denominador==0)
        u=1/pi;
    else
        u=numerador/denominador;
    end

    w=(c^2/(2*pi^3))*u;
end
```

8. Algoritmo que calcula el sinograma del fantasma.

```
function [imagen]=sinograma(objeto,p,q,ang)
imagen=[];
incremento_angulo=ang/p;
angulo=0;
while (angulo<(ang))
    [x,filas]=grafica_proyeccion_objeto(
        objeto,angulo,q);
    angulo=angulo+incremento_angulo;
    imagen=[imagen;filas];
end
end
```

9. Algoritmo directo de Fourier.

```
function [f]=algoritmo_directo(g)
%datos
tamano=size(g);
p=tamano(1);%numero de proyecciones;
q=(tamano(2)-1)/2; %2q+1 es el numero de rayos;
%phantomas;
%[g]=sinograma(elipses,p,q);
g(:,2*q+1)=[];
h=1/q;

%paso 1
G=[];
for j=1:p

    g_fft=(1/sqrt(2*pi))*h*fft(fftshift(g(j,:)));
    G=[G;fftshift(g_fft)];
end
```

D. PROGRAMAS

```
% paso 2

for fila=1:2*q
    v=q*pi-(fila-1)*pi;
    for columna=1:2*q
        u=-q*pi+(columna-1)*pi;
        if ((u^2+v^2)<q^2)
            agamma=(180/pi)*angle(u+i*v);
            agamma=mod(agamma,360);
            j=round(agamma*p/180);
            cercano=(180/p)*j;
            cercano=(pi/180)*mod(cercano,180);
            s=u*cos(cercano)+v*sin(cercano);
            r=round(s/pi);
            k=q+r+1;
            if (k>2*q)
                k=2*q;
            end
            if (k<1)
                k=1;
            end
            j=mod(j,p)+1;
            F(fila,columna)=(1/sqrt(2*pi))*G(j,k);
        else
            F(fila,columna)=0;
        end
    end
end
```

10. Algoritmo de retro-proyección filtrada.

```
function [f]=retroproyeccion(g,angulo)
% Esta función realiza la
% retroproyección filtrada
% p es el número de proyecciones
% el número de rayos va de -q a q
% angulo corresponde al abarcado por las
% proyecciones a partir del eje OY
% ...ambos lados del eje
```

```
% g corresponde al sinograma

%datos
tamano=size(g);
p=tamano(1);
q=floor((tamano(2)-1)/2);
h=1/q;

for j=1:p
    theta1(j)=cos((j-1)*angulo*pi/(180*p));
    theta2(j)=sin((j-1)*angulo*pi/(180*p));
end

% paso 1
for j=1:p
    for k=-q:q
        % convolucion
        v(j,k+q+1)=0;
        for l=-q:q
            v(j,k+q+1)=v(j,k+q+1)+
                RamLak(h,(k-l))*g(j,l+q+1);
        end
        v(j,k+q+1)=v(j,k+q+1)*h;
    end
end

%paso 2
fila=1;
for y=1:-1/128:-1
    columna=1;
    for x=-1:1/128:1

        fbi=0;
        for j=1:p
            s=theta1(j)*x+theta2(j)*y;
            k=floor(s/h);
            u=s/h-k;

            if (k>=q)
                fbi=fbi+v(j,2*q+1);
            end
        end
    end
end
```

D. PROGRAMAS

```
elseif (k<=-q)
    fbi=fbi+v(j,1);
else
    fbi=fbi+(1-u)*v(j,k+q+1)+u*
    v(j,k+1+q+1);
end
end
fbi=(2*pi/p)*fbi;
f(fila,columna)=fbi;
columna=columna+1;
end
fila=fila+1;
end
end
```

11. Programa escrito en Turbo Pascal que realiza la FFT hasta 1024 puntos. Los resultados pueden introducirse por archivo y también por teclado. Las salidas se obtienen en archivo.

```
program FFT_base_2;
uses crt;
const
    Pi=3.1415926535897932;
type
    complejo=record
        p_real:real;
        p_imag:real
    end;
    lista=array[0..1023] of integer;
    vector=array[0..1023] of complejo;

var
    fichero:Text;modulo:real;
    numero_puntos:integer;
    total_fases,fase,mariposas,longitud_mariposa,
    bloque,primero,k:integer;
    numeracion_indices_inicial:lista;
    H,h0:vector;
    producto,suma,resta:complejo;
    W:complejo;
    a,b:complejo;
    respuesta:char;
```

```
procedure datos(var h:vector);
{Obtención de los datos iniciales h0}
var
    k,opcion:integer;
begin
    clrscr;
    writeln('Introducción de los datos
    iniciales');
    writeln('=====
    =====');
    writeln;
    writeln('1. Introducir datos por el
    teclado');
    writeln('2. Introducir datos mediante
    archivo datos.txt');
    writeln;
    write('Elige opción: ');
    readln(opcion);
    if opcion=1
    then begin
        writeln('Introduce los datos as;
        --> parte real    parte imaginaria');
        for k:=0 to numero_puntos-1 do
            readln(h[k].p_real,h[k].p_imag);
        end
    else begin
        assign(fichero,'datos.txt');
        reset(fichero);
        for k:=0 to numero_puntos-1 do
            readln(fichero,h[k].p_real,
            h[k].p_imag);
        close(fichero);
    end;
end;

end;

function potencia(base,exponente:integer)
:integer;
(* Calcula la potencia real *)
var
```

D. PROGRAMAS

```
    resultado,i:integer;
begin
    resultado:=1;
    for i:=1 to exponente do
        resultado:=resultado*base;
    potencia:=resultado;
end;

procedure adiciona(v,w:complejo;var
resultado:complejo);
(* Calcula la suma de dos números
complejos *)
begin
    resultado.p_real:=v.p_real+w.p_real;
    resultado.p_imag:=v.p_imag+w.p_imag;
end;

procedure sustrae(v,w:complejo;var
resultado:complejo);
(* Calcula la suma de dos números
complejos *)
begin
    resultado.p_real:=v.p_real-w.p_real;
    resultado.p_imag:=v.p_imag-w.p_imag;
end;

procedure multiplica(v,w:complejo; var
resultado:complejo);
(* Calcula el producto de dos números
complejos *)
begin
    resultado.p_real:=v.p_real*w.p_real
        -v.p_imag*w.p_imag;
    resultado.p_imag:=v.p_real*w.p_imag
        +v.p_imag*w.p_real;
end;

procedure Buneman(var indices:lista);
var
    etapas,k,L,L1,L2:integer;
begin
```

```
    etapas:=round(Ln(numero_puntos)/
    Ln(2));
    for k:=1 to etapas do
    begin
        for L:= 1 to potencia(2,k-1) do
        begin
            L1:=L-1;
            L2:=L1+potencia(2,k-1);
            indices[L1]:=2*indices[L1];
            indices[L2]:=indices[L1]+1;
        end;
    end;
end;

procedure primera_etapa(var h:vector;h0:
vector;indices:lista);
var
    k:integer;
begin
    for k:=0 to numero_puntos-1 do begin
        h[k].p_real:=h0[indices[k]].p_real;
        h[k].p_imag:=h0[indices[k]].p_imag;
    end;
end;

procedure muestra_DFT(h:vector);
var
    k:integer;
begin
    for k:=0 to numero_puntos-1 do begin
        write(h[k].p_real,' ',h[k].p_imag);
        writeln;
    end;
end;

begin
    (* Menú Principal *)
    clrscr;
    write('¿Cuántos Puntos hay que
```

D. PROGRAMAS

```
    computar? ');
readln(numero_puntos);
datos(h0); (* Recogida de datos de
partida *)
muestra_DFT(h0);
Buneman(numeracion_indices_inicial);
primera_etapa(H,h0,
    numeracion_indices_inicial);
    (* Asigna primera DFT de 1 Pto *)
total_fases:=Round(Ln(numero_puntos)
    /Ln(2));
for fase:=1 to total_fases do begin

    mariposas:=numero_puntos div
        potencia(2,fase);
    longitud_mariposa:=potencia(2,fase);

    for bloque:=1 to mariposas do begin
        primero:=(bloque-1)*longitud_
            mariposa;

        for k:= primero to (primero+
            (longitud_mariposa div 2)-1)
            do begin
                W.p_real:=Cos(-2*Pi*
                    (k-primero)/longitud_mariposa);
                W.p_imag:=Sin(-2*Pi*
                    (k-primero)/longitud_mariposa);

                a.p_real:=H[k].p_real;
                b.p_real:=H[k+(longitud_
                    mariposa div 2)].p_real;
                a.p_imag:=H[k].p_imag;
                b.p_imag:=H[k+(longitud_
                    mariposa div 2)].p_imag;
                multiplica(W,b,producto);
                adiciona(a,producto,suma);
                sustrae(a,producto,resta);

                H[k]:=suma;
                H[k+(longitud_mariposa div
```



```
        2)]:=resta;
    end;

    end;

end;
writeln('==== Resultado FFT ====')
;writeln;
muestra_DFT(H);
assign(fichero,'DFTdatos.txt');
Rewrite(fichero);
for k:=0 to numero_puntos-1 do
    writeln(fichero,H[k].p_real,' ',
        H[k].p_imag);
close(fichero);

assign(fichero,'DFTModul.txt');
Rewrite(fichero);
for k:=0 to numero_puntos-1 do
    begin
        modulo:= H[k].p_real*
            H[k].p_real+H[k].p_imag*H[k].p_imag;
        modulo:=Sqrt(modulo)/numero_puntos;
        writeln(fichero,modulo);
    end;
close(fichero);

writeln('Pulsa Intro para cerrar');
readln;

end.
```