UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CÁLCULO CIENTÍFICO (6109)
SEMESTRE II-2013

PROYECTO: Análisis Espectral

Introducción

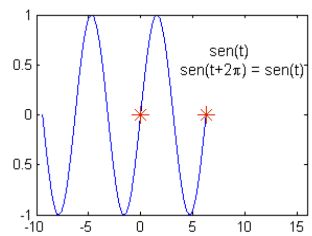
Sabemos que si se poseen $n+1$ evaluaciones de una cierta función f en $n+1$ puntos distintos (nodos de interpolación) es posible encontrar un polinomio p_n de grado menor igual a n tal que p interpole a f. Específicamente si $x_0, x_1, \cdots, x_n$ son los nodos de interpolación, todos distintos entre sí, entonces existe un único $p_n \in \Pi_n$ tal que

$$p_n(x_i) = f(x_i) = y_i, \tag{1}$$

donde Π_n representa el espacio vectorial de todos los polinomios de grado menor o igual a n. Si B_n es una base de Π_n, denotada por $B_n = \{b_0, b_1, \cdots, b_n\}$, entonces *el polinomio interpolador* p_n posee la siguiente forma

$$f(x) \approx p_n(x) = \sum_{k=0}^{n} c_k b_k(x), \tag{2}$$

donde los coeficientes c_k de la combinación lineal (2) que define a p_n se obtienen imponiendo las llamadas n *condiciones de interpolación* descritas en (1).



Ahora bien, por un momento imagine que la función f posee cierta periodicidad en su comportamiento, es decir, f está asociada a algún fenómeno de naturaleza cíclica (sonidos musicales, trayectorias de cuerpos celestes, etc). En estos casos es claro que aproximar a f mediante un elemento de Π_n (polinomio) no resulta la mejor elección. De hecho resulta mucho más conveniente escribir f como una combinación lineal de senos y cosenos, es decir, utilizar *interpolación trigonométrica* en vez de *interpolación polinomial*. En este nuevo contexto, decimos que la f es una función periódica, si existe una constante $p > 0$, que denominaremos *período*, tal que $f(t) = f(t+p)$ para todo t. Por ejemplo, la función seno es periódica con $p = 2\pi$.

Serie de Fourier

Si $\mathcal{B} = \{\varphi_0, \varphi_1, \cdots, \varphi_{n-1}\}$ es una base ortonormal de un cierto espacio vectorial $\mathcal{V}$ con producto interno $\langle ., . \rangle$, entonces cada $x \in \mathcal{V}$ se puede expresar como

$$x = \langle \varphi_0, x \rangle \varphi_0 + \langle \varphi_1, x \rangle \varphi_1 + \cdots + \langle \varphi_{n-1}, x \rangle \varphi_{n-1}.$$ [1]

Esta última expresión recibe el nombre de *Expansión de Fourier* y los escalares $c_j = \langle \varphi_j, x \rangle$ reciben el nombre de *Coeficientes de Fourier*.

Suponga que $\mathcal{V}$ es el espacio de las funciones periódicas con $p = 2\pi$. En este caso, dependiendo de las características de los elementos de $\mathcal{V}$ es posible definir dos productos internos sobre $\mathcal{V}$:

1. **Caso Continuo:** Cuando $f, g \in \mathcal{V}$ son continuas se tiene que

$$\langle f, g \rangle = \int_{-\pi}^{\pi} f(t)\overline{g(t)}dt, \tag{3}$$

donde $\overline{g(t)}$ denota al transpuesto conjugado de $g(t)$.

[1] El proceso de Gram Schmidt construye una base ortonornal a partir de una que no lo es!!

2. **Caso Discreto:** Cuando solamente se conocen evaluaciones de la función f en ciertos puntos equidistantes x_k con $k = 0, 1, \cdots, n-1$ definidos como $x_k = 2\pi k/n$. En este caso se tiene que

$$\langle f, g \rangle_n = \sum_{k=0}^{n-1} f(x_k)\overline{g(x_k)}. \tag{4}$$

Es claro que, para el proceso de interpolación, nos conviene el caso discreto pues sólo poseemos evaluaciones de la función f.

Interpolación trigonométrica

Para hacer más comprensible la lectura, no pierda de vista que simplemente queremos encontrar una expresión similar a (2) pero usando funciones trigonométricas en vez de polinomios. Para ello considere:

- Al conjunto de las funciones periódicas de las cuales se pueden conocer n evaluaciones de las mismas en los puntos $x_k = 2\pi k/n$ con $k = 0, 1, \cdots, n-1$. Denotemos a este conjunto por $\mathcal{C}$.
- La función de producto interno discreto definida en (4).
- Al conjunto $\mathcal{B} \subset \mathcal{C}$ definido como $\mathcal{B} = \{\phi_0, \phi_1, \cdots, \phi_{n-1}\}$ donde $\phi_k = \phi_k(x) = e^{ikx}$ y la variable i representa la unidad compleja, es decir, $i = \sqrt{-1}$. **Este conjunto $\mathcal{B}$, usando el producto interno $\langle ., . \rangle_n$, es un conjunto ortogonal en el sentido que**

$$\langle \phi_j, \phi_k \rangle_n = \begin{cases} n & \text{si } (j-k)/n \text{ es un valor entero} \\ 0 & \text{en otros casos} \end{cases}$$

Bajo este escenario, se tiene que la siguiente función p es el **interpolador trigonométrico** de la función f en los nodos $x_k = 2\pi k/n$

$$f(x) \approx p(x) = \sum_{j=0}^{n-1} c_j \phi_j = \sum_{j=0}^{n-1} c_j e^{ijx}, \tag{5}$$

donde

$$c_j = \frac{\langle f, \phi_j \rangle_n}{\langle \phi_j, \phi_j \rangle_n} = \frac{1}{n} \sum_{k=0}^{n-1} f(x_k)\overline{\phi_j(x_k)} = \frac{1}{n} \sum_{k=0}^{n-1} f(x_k) e^{-ij x_k}, \quad j = 0, 1, \cdots, n-1 \tag{6}$$

pues si $\phi_j = e^{ijx}$ entonces $\overline{\phi_j} = e^{-ijx}$. Los c_j definidos por (6) se conocen como *Coeficientes de Fourier.*

Importante

Seguramente se está preguntando por qué se dice que la interpolación de la función f se realizó mediante senos y cosenos, si en la expresión (5) no aparecen funciones trigonométricas sino exponenciales. La respuesta es que SI aparecen pero de *forma implícita.* Basta recordar la famosa **Fórmula de Euler**:

$$e^{i\theta} = \cos\theta + i \sin\theta, \tag{7}$$

y recuerde que ϕ_k en (5) se define como $\phi_k = e^{ikx}$.

Como dato curioso, observe que al sustituir θ por π en (7) se tiene que $\cos\pi = -1$ y $\sin\pi = 0$ y por tanto

$$e^{i\pi} + 1 = 0$$

que es conocida como una de las identidades más hermosas de la matemática. Su belleza radica en su simplicidad y en relacionar 5 números fundamentales de la matemática: e, π, i, 1 y el 0.

Transformada Discreta de Fourier

Antes de continuar haremos algunas consideraciones y ciertos cambios de variable para simplificar la notación:

- Recuerde que (5) es el interpolador de f en los nodos $x_k = 2\pi k/n$, por lo tanto sea $y_k = p(x_k) = f(x_k)$.
- Analicemos por un momento el valor de e^{-ijx} evaluado en los nodos $x_k = 2\pi k/n$:

$$e^{-ij{\color{red}x_k}} = e^{\frac{-ij2\pi k}{n}} = e^{\frac{-2\pi i}{n}jk} = \left(e^{\frac{-2\pi i}{n}}\right)^{jk} = \omega^{jk},$$

  con $\omega = e^{\frac{-2\pi i}{n}}$.
- Con la finalidad de adaptarnos a la notación empleada por Matlab, obviaremos el escalamiento de $1/n$ en (6) para obtener que

$$c_j = \sum_{k=0}^{n-1} f(x_k)e^{-ij{\color{red}x_k}} = \sum_{k=0}^{n-1} y_k\omega^{jk}, \quad j = 0, 1, \cdots, n-1 \tag{8}$$

- Al evaluar (5) en $x = x_k$ y al restituir el escalamiento de $1/n$ se obtiene que

$$y_k = f(x_k) = \frac{1}{n}\sum_{j=0}^{n-1} c_j e^{ij{\color{red}x_k}} = \frac{1}{n}\sum_{j=0}^{n-1} c_j\overline{\omega}^{jk}, \quad k = 0, 1, \cdots, n-1 \tag{9}$$

  pues recuerde que $\overline{\omega} = e^{\frac{2\pi i}{n}}$.

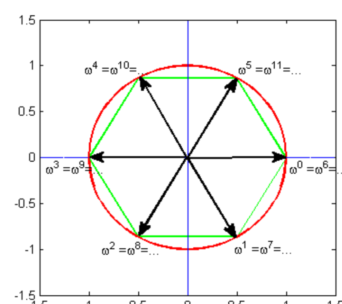
Considere las secuencias $\{c_j\}$ y $\{y_k\}$ generadas por (8) y (9) respectivamente. La secuencia $\{c_j\}$ se conoce con el nombre de *Transformada Discreta de Fourier* (TDF) de $\{y_k\}$, mientras que $\{y_k\}$ se conoce como la *Inversa de la Transformada Discreta de Fourier* (inv TDF) de la secuencia $\{c_j\}$.

Algoritmo Transformada Rápida de Fourier

Para hallar cada c_j se requiere de n multiplicaciones y $n-1$ sumas en aritmética compleja. Por lo tanto, calcular los n coeficientes de Fourier es un proceso de aproximadamente n^2 operaciones complejas. El algoritmo de la *Transformada Rápida de Fourier*, FFT por sus siglas en inglés[2], es tan famoso porque es capaz de calcular dichos coeficientes en un tiempo aproximado de $n\log_2 n$ cuando n es potencia de 2. Observe la Tabla para comprender la magnitud de la reducción.

n	$n\log_2 n$	n^2
$2^7 = 128$	896	16384
$2^8 = 256$	2048	65536
$2^9 = 512$	4608	262144
$2^{10} = 1024$	10240	1048576

No describiremos en detalle el algoritmo de la Transformada Rápida de Fourier, pero podemos ilustrar con un ejemplo simple donde radica la rapidez de dicho algoritmo. Considere el polinomio $p(z) = z^n - 1$. Las n raíces de este polinomio vienen dadas por $\{\omega^0, \omega^1, \omega^2, \cdots, \omega^{n-1}\}$ donde ω, tal y como lo hemos definido anteriormente, es igual a $e^{\frac{-2\pi i}{n}}$. Estas raíces son cíclicas en el sentido que si $k \geq n$ entonces $\omega^k = \omega^{k \bmod n}$. Por ejemplo, para $n = 6$ podemos observar en la Figura donde se ubica cada ω^k, específicamente se tiene que



$\omega^0 = \omega^6 = \cdots$	$\omega^1 = \omega^7 = \cdots$	$\omega^2 = \omega^8 = \cdots$
$\omega^3 = \omega^9 = \cdots$	$\omega^4 = \omega^{10} = \cdots$	$\omega^5 = \omega^{11} = \cdots$

Ahora bien, recuerde que para construir los c_j se requieren muchas potencios de ω pero, como acabamos de ver, la repetición de estas potencias nos permite calcularlas de forma inteligente para evitar cálculos innecesarios. Además observe que $\omega^0 = 1$ y $\omega^3 = -1$.

[2]Fast Fourier Transform

Proyecto: Análisis espectral

Otra utilidad importante de la Transformada Discreta de Fourier es que permite identificar frecuencias básicas en un conjunto de datos, este proceso se conoce como *análisis espectral* sobre datos discretos. En este proyecto utilizaremos la TDF para realizar un simple análisis espectral que nos permita:

1. Dado un pulso sobre un teclado telefónico, codificar en un señal sonora el tono asociado al dígito pulsado.

2. Dado un señal sonora asociada a un número telefónico de 11 dígitos, decodificar cuáles fueron los dígitos marcados.

El código anexo `tonos.m`, realiza el proceso de codificación. El objetivo central del proyecto es que ustedes realicen el proceso de decodificación (segundo ítem). Pasemos a describir en detalle el proceso de codificación.

Codificación

Al marcar un número telefónico el mismo es codificado en un señal sonora siguiendo el sistema de *Tono Dual Multi-Frecuencia* (TDMF). En este proyecto cada señal tendrá una duración de 0.25 segundos. El sistema de codificación TDMF consiste en lo siguiente:

A cada fila y columna del teclado de un teléfono se le asigna una frecuencia básica, así el vector denotado por f_r contiene las frecuencias para las 4 filas, mientras que f_c contiene las frecuencias para las 3 columnas. Estos vectores se definen como

$$\begin{aligned} f_r &= [697 \quad 770 \quad 852 \quad 941] \\ f_c &= [1209 \quad 1336 \quad 1477] \end{aligned}$$

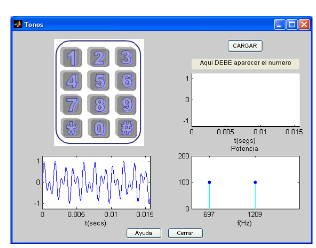
La señal y asociada a la tecla ubicada en la posición (k, j) se genera mediante la superposición de dos tonos fundamentales con frecuencias $f_r(k)$ y $f_c(j)$ respectivamente, es decir, $y = (y_1 + y_2)/2$ donde

$$y_1 = \sin(2\pi f_r(k)t) \quad y \quad y_2 = \sin(2\pi f_c(j)t),$$

siendo $t \in [0, 0.25]$ una variable discreta que representa al tiempo de duración de la señal, es decir, t es un vector de componentes reales donde definidas como $t_{k+1} = \frac{k}{F_s}$, con $k = 0, 1, \cdots, n-1$ y F_s es una constante conocida como el *sampling rate*. El proceso anteriormente descrito se resume en las siguientes instrucciones:

```
Fs = 32768;
t = 0:1/Fs:0.25;
y1 = sin(2*pi*fr(k)*t);
y2 = sin(2*pi*fc(j)*t);
y = (y1 + y2)/2;
n = length(y); % n=8193
sound(y,Fs) % Produce un sonido de 0.25 seg de duracion, asociado
            % al digito ubicado en la posicion (j,k) del teclado
```

La Figura de la derecha presenta la interfaz del programa `tonos.m`. Al pulsar un dígito en el teclado, digamos el número 1, se genera el sonido de la señal asociada y (instrucción `sound(y,Fs)`), además se despliegan dos gráficas: En la esquina inferior izquierda se puede observar la gráfica de la señal y cuando la variable t está en el intervalo $[0, 0.015]$. Por otro lado, en la esquina inferior derecha, se muestran las frecuencias básicas asociadas al dígito pulsado, que en este caso por tratarse del número uno se observa que las frecuencias son 697 por la fila y 1209 por la columna.



Decodificación

En este punto conviene analizar la TDF desde el punto de vista del análisis espectral. Recordemos que la TDF viene dada por la expresión

$$c_j = \sum_{k=0}^{n-1} y_k \omega^{jk}, \quad j = 0, 1, \cdots, n-1.$$

Cada coeficiente de Fourier está asociado a un frecuencia en particular en el siguiente sentido: Definamos a la frecuencia f_j, como

$$f_j = j\frac{F_s}{n}, \quad j = 0, 1, \cdots n-1.$$

El valor absoluto de c_j define la magnitud de la frecuencia f_j presente en la señal y. Recuerde que $y = (y_0, y_1, \cdots, y_{n-1})^t$. Por lo tanto, para un dígito en específico, digamos el número uno, se espera que el valor absoluto de los dos coeficientes de Fourier asociados a las frecuencias 697 y 1209 sean valores altos en comparación a las magnitudes del resto de los c_j.

Ayuda: Descomente las líneas en `tonos.m` identificadas como `AYUDA PARA LA DECODIFICACION` y observe la nueva gráfica que aparece en la esquina inferior derecha!!

Datos de entrada

Archivo `signal.mat`, donde se encuentra almacenada una señal y de 11 dígitos. La señal asociada a cada dígito tiene un tamaño de 8192 elementos y 0.25 segundos de duración. Por lo tanto, el vector y posee en total 90112 elementos.

Requerimientos prácticos

Al pulsar el botón `CARGAR` el programa debe cargar la señal almacenada en el archivo de entrada `signal.mat` y decodificar dicha señal. Específicamente el programa DEBE

1. Reproducir la señal cargada.
2. Decodificar la señal cargada y colocar los dígitos en marcados en la señal en el *textbox* indicado en la interfaz de *tonos.m*
3. Graficar la señal cargada en la esquina superior derecha de *tonos.m*

Requerimientos teóricos

1. Dado un vector $y \in \mathbb{C}^n$, determine a la matriz F tal que $Fy = c$ donde $c = (c_0, c_1, \cdots c_j, \cdots, c_{n-1})^t$, y c_j se genera mediante (8). La matriz F se conoce como **matriz de Fourier**.
2. Demuestre que $F^H F = nI$, donde I representa la identidad de orden n y F^H es la transpuesta conjugada de F, es decir, si $F = [f_{kj}]$ entonces $F^H = [\overline{f}_{jk}]$, siendo $\overline{f}_{jk}$ el transpuesto conjugado de f_{jk}.
3. Observe que del ítem anterior se desprende que $F^{-1} = \frac{1}{n}F^H$. Siguiendo la notación del primer ítem, demuestre que (9) se puede escribir matricialmente como $y = F^{-1}c$ donde $y = (y_0, y_1, \cdots y_k, \cdots, y_{n-1})^t$

Consideraciones finales

1. El proyecto debe ser realizado por dos personas máximo.
2. La implementación debe ser en Matlab u Octave.

3. Usted deberá entregar en la fecha prevista UNICAMENTE un archivo en formato zip (`apellido1apellido2.zip`), que contenga como mínimo dos archivos: `informe.pdf` y `tonos.m`[3]

4. El informe electrónico (formato `pdf`) debe cumplir con los siguientes aspectos:

   - Interlineado 1.5, letra tamaño 12. El número de páginas de CONTENIDO no debe ser menor a 3 ni mayor a 6.
   - Solución de todos y cada uno de los requerimientos del proyecto. Para los requerimientos prácticos debe describir su estrategia de solución.

5. Se tomarán en cuenta aspectos concernientes a la programación: eficiencia, facilidad de uso, etc.

6. Fecha de entrega: ~~21/03/2014~~ 19/05/2014 (vía correo electrónico a: `victorjfn@gmail.com` o `samuelnacache@gmail.com` CON COPIA A `mmonsalv@gmail.com` hasta las 11:59pm).

Grupo Docente de Cálculo Científico I / Realizado por M. Monsalve

[3] LÉASE BIEN: el informe DEBE estar en formato PDF.