VNiVERSiDAD
D SALAMANCA

Tema 4: Modelización de datos y técnicas de Fourier

Series Temporales: Análisis en el dominio del tiempo

Esta primera sesión tiene por objetivo el análisis de series temporales en el dominio del tiempo. Se introduce el concepto de función de autocorrelación y se procede a su cálculo en casos concretos.

Series Temporales: Análisis en el dominio del tiempo

Descripción del trabajo que se realizará en la Sesión 4.1

Los objetivos básicos de esta primera sesión del tema se centran en el diseño por cada estudiante de un *set* de programas para la lectura, manipulación de series temporales largas y tratamiento en el **dominio del tiempo**. Se programa en Fortran compatible con FORTRAN 77 cuya correcta implementación se hace mediante comparación de los resultados obtenidos con analíticos y/o numéricos disponibles.

Generador de números aleatorios: El mismo usado en la Parte 3. No se usa semilla.

Longitud de las series temporales: Se exige generar código con matrices de longitud variable. Se recomienda usar el comando *PARAMETER* para este fin. Aunque en Fortran 95 es posible usar dimensionamiento dinámico, se usará fijo de hasta 10 millones.

Precisión: Doble.

Formato: En principio, libre. El uso de subrutinas está prohibido.

Depurador (*debugger*): Aunque al nivel de complejidad demandado por la aplicación el uso de un *debugger* se recomienda aprender el uso del depurador *gdb* [1].

Compilación: La usada hasta ahora. Si se prevé usar el *debugger* es necesario compilar con la opción –*g*, si se desea compilar varios ficheros *.f* (situados en el mismo directorio) para generar un ejecutable basta con usar el comodín * (por ejemplo, *.*f*).

Declaración de variables: Aunque FORTRAN permite el uso de variables no declaradas, es buena práctica el uso de *IMPLICIT NONE*.

Documentación. Es obligatorio documentar el código definiendo todas las variables e identificando la función de los bloques principales.

Códigos realizados en la primera sesión

Código 1: Generación y operaciones básicas sobre una serie temporal

Casos de aplicación

1 – Creación de un programa principal con llamada a un generador de números aleatorios de distribución uniforme en el rango [0,1]. Generación en formato libre en una matriz declarada como real designada como *matrix_gen* de una serie de un millón de números aleatorios. Escritura en un

fichero llamado "*serie.txt*" de la serie generada (úsese la unidad 15). Apertura del fichero "*serie.txt*" usando la unidad 16 y lectura de la serie en una matriz (*matrix_read*) declarada como real.

2 – Añadir los cálculos siguientes: Valor esperado (X) y segundo momento central (μ_2). ¿Qué miden estas magnitudes en una función de distribución?

Recomendaciones en el trabajo con matrices: Debemos asegurarnos de no acceder a posiciones no existentes de una matriz. Como consejo, debe utilizarse la compilación con "**bound-checking**" (*-fcheck-bounds*) [2]; búsquese en el manual la opción que lo permite y úsese siempre en la sesión. Por otra parte, dado que los compiladores modernos no abortan la ejecución si surgen excepciones, para evitar que se enmascaren errores numéricos es muy recomendable incluir la opción de compilación *-ffpe-trap=zero,invalid,overflow,underflow* [3].

Código 2: Generación de la serie temporal: Función de autocorrelación

Se recuerda que dada una función muestra *x(t)* (por ejemplo, una parte finita de una serie temporal infinita) son de especial interés su valor medio y la función temporal de autocorrelación de la función muestra:

$$\begin{aligned} \overline{x} &= A\left[x(t)\right] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)dt \\ \Re_{xx}(\tau) &= A\left[x(t)x(t+\tau)\right] = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)x(t+\tau)dt \end{aligned} \tag{1}$$

La función de autocorrelación tiene las siguientes propiedades de las que se pueden extraer test simples que debe cumplir en tiempo de ejecución el código generado si se usa para estudiar un sistema ergódico:

$$\left|\Re_{xx}(\tau)\right| \le \Re_{xx}(0) \tag{2}$$

$$\Re_{xx}(-\tau) = \Re_{xx}(\tau) \tag{3}$$

$$\Re_{xx}(0) = E\left[X^2(t)\right] \tag{4}$$

$$\lim_{|\tau|\to\infty} \Re_{xx}(\tau) = 0 \tag{5}$$

Códigos y aplicaciones

1– **Cálculo de la función de correlación de un generador uniforme de números aleatorios**. En el apartado anterior se creó de un programa principal con llamada a un generador de números aleatorios de distribución uniforme en el rango [0,1) que se usó para generar un millón de números aleatorios. Debe mantenerse la estructura del programa[1], pero debe modificarse el programa original para incluir una columna en la que se mida el tiempo. Se sugiere que el paso temporal (*timestep*) sea una variable del programa. A efectos de test, el contenido de la columna son los números naturales del índice; su valor real se obtiene multiplicando el índice por la variable *timestep*.

Téngase presente que en el cálculo de la correlación no siempre es necesario emplear un millón de eventos (¿por qué?), también debemos asegurarnos de no acceder a posiciones no existentes de una matriz. Debe usarse obligatoriamente la compilación con *"bound checking"* que permite abortar la ejecución si hay errores. Léase el manual del compilador.

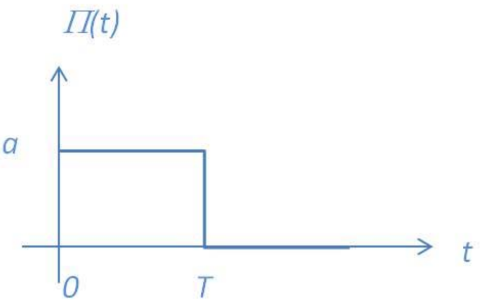
Escríbase el código, compílese y ejecútese usando una sección de una serie temporal generada mediante un generador de números aleatorios de distribución uniforme en el rango [0,1) con longitudes de correlación razonables. ¿Por qué el cálculo de la autocorrelación sirve como criterio para determinar la calidad de un generador de números aleatorios?

2 **Cálculo de la función de correlación de un generador uniforme de números aleatorios de valor promedio nulo**. Repítase el caso anterior para una serie temporal de números aleatorios uniforme de valor esperado nulo.

3 - **Calcúlese numérica y analíticamente la autocorrelación de la función puerta, *Π(t)*:**

Esta función se denomina *Π* ó función puerta y modeliza eventos con aparición durante un tiempo limitado.

$$\Pi(t) = \begin{cases} a & si \quad 0 \le t \le T \\ 0 & en \quad otro \quad caso \end{cases}$$



A nivel del código debe implementarse una sección que muestree con un *timestep* ajustable por el usuario (realmente, esta variable no se proporciona a voluntad del usuario si no que tiene implicaciones muy serias en la calidad de discretización de la señal[2]). Esta sección servirá para el análisis de señales deterministas y para validar el código.

[1] Escritura en un fichero llamado "*serie.txt*" de la serie generada (úsese la unidad 15). Apertura del fichero "*serie.txt*" usando la unidad 16 y lectura de la serie en una matriz (*matrix_read*) declarada como real.
[2] Cf. Teorema de Shannon que analizaremos en la segunda sesión.

Bibliografía

[1] http://www.gnu.org/software/gdb/documentation/ , http://www.lsi.us.es/~javierj/ssoo_ficheros/GuiaGDB.htm

[2] https://gcc.gnu.org/onlinedocs/gfortran/Code-Gen-Options.html

[3] https://gcc.gnu.org/onlinedocs/gcc-8.4.0/gfortran/Debugging-Options.html