

Código: 15741
Licenciatura: LE
Curso: Indiferente
Línea Curricular:
Cuatrimestre: 1º
Créditos: 6



Facultad de Ciencias Económicas y Empresariales
Ekonomi eta Enpresa Zientzien Fakultatea

Programa de la asignatura

Computación y Métodos Numéricos

Año Académico

2007/2008

Curso Académico

Indiferente

Licenciatura

Economía

Línea Curricular/Tipo de Asignatura

Libre elección

Departamento

**Economía Aplicada III
(Econometría y Estadística)**

Profesores que imparten la asignatura:

1. Araceli Garín Martín.

Objetivos de la asignatura:

Iniciar al alumno en el estudio de algoritmos numéricos y su posterior implementación en un ordenador. Estudiar algoritmos capaces de abordar problemas de ordenación, de resolución de ecuaciones y sistemas de ecuaciones lineales y no lineales o de simulación. Introducir al alumno en un lenguaje de programación (C) que le permita implementar los algoritmos estudiados.

Estructura de temas:

1. Nociones básicas sobre algoritmos.

¿Qué es un algoritmo? Medidas de eficiencia. Complejidad. Complejidad computacional de algoritmos numéricos. Análisis de algoritmos. Límites de complejidad.

2. Nociones sobre arquitectura de ordenadores.

Memoria, canales de entrada/salida, unidad central de proceso o CPU. La metáfora del enano computador. Lenguaje de máquina, lenguajes de ensamblado, lenguajes de alto nivel.

3. Introducción al lenguaje de programación C (I).

Nociones sobre trabajo en el sistema UNIX. Compilado, ejecución y depuración de programas en C. Variables escalares, subindicadas (*arrays*) y constantes. Operadores y funciones standard mas usuales. Orden de precedencia en operadores.

4. Introducción al lenguaje de programación C (II).

Declaraciones. Tipos de datos: carácter, enteros, enumeración, y de coma flotante. Conversiones de tipo. Nociones sobre representación de datos e implicaciones sobre la precisión numérica. Relaciones de recurrencia.

5. Introducción al lenguaje de programación C (III).

Sentencias de control de flujo: for, if, do, while, switch. Ejemplos de uso. Funciones: anatomía y uso. Convenciones sobre el paso de argumentos. Recursión. Funciones definidas en stdio y math.

6. Procesos iterativos: órdenes de convergencia.

Resolución de ecuaciones. Método de bisección simple. Método de Newton-Rapson y de la secante. Comparación de sus respectivos órdenes de convergencia. Generalización a sistemas de ecuaciones. Métodos de Newton-Rapson y Gauss-Seidel.

7. Métodos de ordenación.

Análisis teórico: orden de complejidad óptimo alcanzable para algoritmos "en el sitio". Métodos simples de complejidad $O(n^2)$: comparación, inserción y burbuja. Métodos de complejidad $O(n \log n)$: Quicksort y Heapsort.

8. El lenguaje de programación C (IV).

Direccionamiento indirecto. Punteros: variables que contienen la ubicación en memoria de otras variables. Algebra de punteros. Punteros y funciones: paso de argumentos de referencia haciendo uso de punteros. Algunas funciones revisitadas: funciones que retornan punteros. Ejemplos de utilización de punteros.

9. El lenguaje de programación C (V).

Tipos de datos definidos por el usuario: estructuras. Punteros a estructuras y estructuras ligadas. Listas, pilas, árboles. Ejemplos de uso.

10. Breve repaso de nociones de álgebra lineal.

Solución de sistemas de ecuaciones lineales: método de eliminación gaussiana. Inversión de matrices: método de Gauss-Jordan. Complejidad algorítmica. Estabilidad numérica y su mejora: pivotado total y parcial.

11. Solución de ecuaciones lineales sin inversión de la matriz de coeficientes.

La descomposición de Choleski. La descomposición LU. Sustitución hacia atrás. Complejidad y estabilidad numérica. Cálculo de vectores y valores propios de matrices simétricas. El método de la potencia. El método de Jacobi.

12. Simulación (I). Generación de números aleatorios.

Números aleatorios y Método de Montecarlo. Integración numérica. Otras aplicaciones. Generación de observaciones pseudoaleatorias uniformes. Algoritmos de generación multiplicativos: elección del multiplicador, módulo y semilla.

13. Simulación (II). Generación de variables no uniformes.

Método de inversión. Ejemplos. Método de rechazo. Ejemplos. Método de transformación. Ejemplos.

Competencias específicas de la asignatura:

- C1.- Capacidad de describir y modelizar problemas matemáticos y expresarlos formalmente como problemas numéricos.
- C2.- Capacidad de abstracción.
- C3.- Capacidad de manipular un ordenador mediante su sistema operativo. Crear y hacer funcionar un programa.
- C4.- Capacidad de traducir del lenguaje de las matemáticas al lenguaje de programación.

Metodología docente:

La docencia de la asignatura se plantea de acuerdo con cuatro tipos de actividades. La clase magistral, a desarrollar por el profesor en el aula. En ella se abordará el contenido teórico de la asignatura. Este tipo de clases serán alternadas con la resolución de ejercicios y ejemplos a modo de ilustración de la teoría. Se propondrán al alumno colecciones de ejercicios a resolver en el ordenador, tutorizados por el profesor. Por último serán propuestas tareas evaluadoras a fin de comprobar el avance en el proceso de aprendizaje del alumno.

Sistemas de evaluación:

Se trata de una asignatura donde el componente práctico es fundamental. La evaluación se realizará de manera continua en base a una serie de tareas que el alumno deberá ir realizando a medida que avance en el desarrollo del programa y con ayuda del ordenador.

Referencias bibliográficas básicas :

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison Wesley, Reading, Ma., 1974. (510.5 AHO)

[2] F.J Ceballos. *Curso de Programación con C Microsoft*. Ra-ma, Madrid, 1991. (681.3.068 CEB)

Referencias bibliográficas complementarias:

[3] G. Dalquist and Ake Björck. *Numerical Methods*. Prentice Hall, Englewood Cliffs, N.J., 1974. (519.6 DAH)

[4] L. Devroye. *Non-uniform random variable generation*. Springer-Verlag, 1986.

[5] G.S. Fishman. *Conceptos y Métodos en la Simulación Digital de Eventos Discretos*. Limusa, México, 1978.

[6] B.W. Kernighan and D.M. Ritchie. *The C Programming Language*. Prentice-Hall, New Jersey, 1978. (681.3.068 KER)

[10] L. Kronsjo. *Algorithms. Their Complexity and Efficiency*. John Wiley and Sons, London, 1987. (510.5 KRO)