



UNIVERSIDAD DE MURCIA

DEPARTAMENTO DE MATEMÁTICAS

MÉTODOS NUMÉRICOS. PRÁCTICA 9

Ésta es la práctica que corresponde al subproyecto `ecuacionesPolinomicas`. Hemos añadido al paquete `auxiliar` la clase `Polinomio.java` y un nuevo paquete `ORG.netlib.math.complex`. La clase `Polinomio.java` contiene distintos métodos para construir polinomios, operar con ellos y aproximar raíces, y llama a la clase `Complex.java` del paquete `ORG.netlib.math.complex`, que permite trabajar con números complejos. Los ejercicios de esta práctica estarán contenidos en el paquete `ecuacionesPolinomicas.practica9`.

1. Considérense los polinomios

$$p(x) = x^7 - 45x^5 + 33x^3 - 2x^2 + x - 1$$

y

$$q(x) = x^3 - 2x^2 - x + 1.$$

- Calcular $p(x) \cdot q(x)$.
 - Calcular el cociente $c(x)$ y el resto $r(x)$ en la división de $p(x)$ entre $q(x)$.
 - Comprobar que $p(x) = c(x) \cdot q(x) + r(x)$.
 - Añadir el método `resta` a la clase `Polinomio` y calcular $(p(x) - r(x)) : q(x)$.
2. Añadir a la clase `Polinomio` un método `taylor` que construya el polinomio de Taylor de un polinomio centrado en un número x_0 (real o complejo). Comprobar su funcionamiento con los polinomios del ejercicio 1, construyendo sus polinomios de Taylor centrados en $x_0 = 1, 0.5$ y 2 .

3. Escribir un método

```
Polinomio determinanteMenores(Polinomio[] [] p)
```

que calcule el determinante de una matriz cuadrada de polinomios y añadirlo a la clase `Matrices`. Usarlo para diseñar un método

```
Polinomio polinomioCaracteristico(double[] [] A)
```

que calcule el polinomio característico de una matriz A .

Considérese la matriz cuadrada tridiagonal de dimensión n

$$A_n = \begin{pmatrix} 0 & 1 & & & & & \\ 1 & 0 & 1 & & & & \\ & 1 & 0 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & 1 & 0 & 1 \\ & & & & & 1 & 0 \end{pmatrix}.$$

Usando el método de Newton para resolver la ecuación característica, calcular sus valores propios para $n = 2, 3, \dots, 10$. ¿Qué ocurre desde $n = 11$ en adelante? Comparar los resultados con los proporcionados por el método de Jacobi.

- Implementar un método `wilkinson(int n)` que devuelva el polinomio de Wilkinson de grado n , $W_n(x) = (x - 1)(x - 2) \cdots (x - n)$. Buscar sus raíces $n - 4, n - 3$ y $n - 2$ para $n = 5, 10, 15, 20$ utilizando en cada caso los métodos de Newton y Laguerre.
- Resolver completamente el ejercicio 20 de la hoja de problemas 4 de Métodos Numéricos.
- Usando el número del DNI constrúyase el polinomio

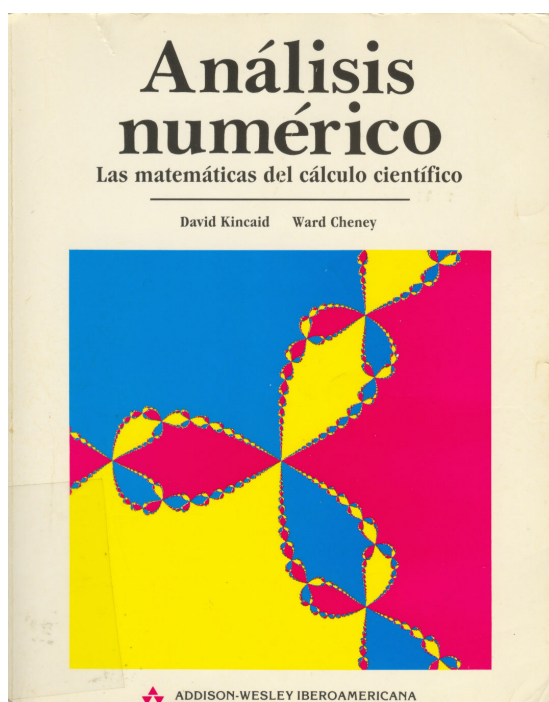
$$p(x) = \sum_{i=0}^9 a[i]x^k$$

con $a[k] = (-1)^{\lfloor \text{DNI}/(k+1) \rfloor} * (\text{DNI} * (k+1) \% \pi)$, donde $\lfloor b \rfloor$ es la parte entera de b y $b \% c$ es el resto de dividir b entre c .

Aproximar todas sus raíces guardándolas en dos listas `raizReal[]` y `raizComplex[]` y comprobar la bondad de las aproximaciones reconstruyendo el polinomio

$$p(x) = a[9] \prod_i (x - \text{raizReal}[i]) \prod_j (x - \text{raizComplex}[j]).$$

7. El objetivo de este ejercicio es crear el dibujo que sale en la portada del libro de Kincaid-Cheney:



La manera de hacerlo viene explicada en la página 108 de dicho libro e involucra el uso del método de Newton con valores complejos. Para poder realizar el dibujo es necesario saber pintar puntos en una ventana de JAVA. Como ilustración se han incluido en el paquete `auxiliar` el fichero

`PanelDibujoSimplificado.java`

y en el paquete `ecuacionesPolinomicas.practica9` el fichero

`PuntosAleatorios.java`.

`PuntosAleatorios.java` es una aplicación que usa el objeto gráfico `PanelDibujoSimplificado.java` para abrir una ventana y llenarla de puntos de colores verde, rojo y azul aleatoriamente escogidos.

Se dará el nombre de `ConjuntoDeJulia.java` a la aplicación que realiza el dibujo pedido; también se guardará en `ecuacionesPolinomicas.practica9`.