Tarea 4 - Problema 1 - Programación avanzada

Nicolás Aylwin

Buscamos generar los Polinomios de Legendre a través del algoritmo de Gram-Schmidt. Estos se caracterizan por ser ortogonales entre sí en el intervalo $[-1,1]$, y porque su norma al cuadrado (o equivalentemente, el producto interno con sigo mismo) es:

$$\langle P_l(x), P_l(x)\rangle = ||P_l(x)||^2 = \frac{2}{2l+1} \tag{1}$$

Con l como el grado del polinomio. Además el producto interno entre dos polinomios se define como:

$$\langle p(x), q(x)\rangle = \int_{-1}^{1} p(x)q(x)dx \tag{2}$$

Por otro lado, el algoritmo de Gram-Schmidt se basa en la obtención de vectores ortogonales a otros vectores. Se representa como (siendo u_i cada vector generado por el proceso y v_i un vector cualquiera):

$$\begin{aligned} u_1 &= v_1 \\ u_2 &= v_2 - proj_{u_1}(v_2) \\ u_3 &= v_3 - proj_{u_1}(v_3) - proj_{u_2}(v_3) \end{aligned}$$

Se continua con este patrón para obtener todos los vectores ortogonales entre sí que uno quiera. Además se tiene que:

$$proj_u(v) = \frac{\langle u, v\rangle}{\langle u, u\rangle} u \tag{3}$$

Si tenemos los dos primeros Polinomios de Legendre, respectivamente $P_0(x) = 1$ y $P_1(x) = x$, podemos usar el algoritmo de Gram-Schmidt comenzando de un polinomio genérico del siguiente grado, como $p(x) = x^2$, y aplicar el algoritmo con el producto interno definido como en Ec.(2). Supongamos obtenemos un polinomio Q_2 que definitivamente es ortogonal a P_0 y P_1, pero en general no cumplirá la Ec.(1). Necesitamos normalizar. Sea para esto c una constante cuqluiera. Podemos imponer que:

$$c\langle Q_2, Q_2\rangle = c\int_{-1}^{1} Q_2Q_2dx = \int_{-1}^{1} (\sqrt{c}Q_2)(\sqrt{c}Q_2)dx = \frac{2}{2\cdot 2+1}$$

Así de forma general se tiene que $\sqrt{c}Q_l(x) = P_l(x)$ con

$$c = \frac{2}{2l+1}\frac{1}{\langle Q_l, Q_l\rangle}$$

Siendo Q_l el polinomio obtenido con el algoritmo de grado l y P_l el polinomio de Legendre de tal grado.

Dejando claro esto, vamos al programa:

Los archivos que permiten esto son 'poli.h' y 'problema1.cc'. El primero es una clase de templates de polinomios, bastante genérica, lo único con sentido mencionar es un constructor que recibe el grado del polinomio y un vector. Este se espera en orden de mayor a menor grado, y de no tener los elementos suficientes rellena con ceros. Esto es particularmente útil para definir polinomios como $x^{cualquiergrado}$ rápidamente, y es el constructor usado al crear cualquier polinomio. El segundo programa es el que en sí realiza lo pedido.

Se comienza en este definiendo la función 'prod_interno' como el producto interno entre dos polinomios (entregados const y por referencia). Se define tal como se expresa en la Ec.(2). Luego la función 'proy' como proyección de un polinomio sobre otro.

Despues se pasa directo al main. En este se comienza definiendo una variable double 'c' para normalizar, un vector de double con solo un uno ('v') y los dos primeros polinomios de legendre (P_0 y P_1) usando 'v'. Se crea también un vector de polinomios llamado 'w' y se almacenan P_0 y P_1 en este. Luego se da un mensaje sobre los polinomios encontrados.

Ahora comienza el ciclo for importante. De 2 a 20 iterando con 'i', comienza creando un polinomio de double de grado 'i' genérico (x^i), llamado 'p_i'. También se crea un polinomio 'acum'. Luego se entra a otro for que itera sobre el vector de polinomios 'w'. Aquí se acumula en 'acum' el resultado de la proyección de 'p_i' sobre todos los polinomios almacenados en 'w' (para asegurarnos que sea ortogonal a todos). Luego se sale de este ciclo y se resta 'acum' a 'p_i', asignando este valor a 'p_i' (ahora es el 'Q_l' mencionado antes). Seguido se normaliza usando c y considerando $i = l$. Por último se almacena este nuevo Polinomio de Legendre en 'w', se imprime a pantalla y se repite el ciclo.

Luego de imprimir todos se ponen a mano (uf) todos los Polinomios de Legendre según mathematica del grado 2 al 20. Seguido se imprime a pantalla el error entre los de grados equivalentes (obtenidos y tabulados) según:

$$\int_{-1}^{1} (P_l(x) - p_l(x))^2$$

Con P_l el Polinomio de Legendre obtenido y p_l el entregado por mathematica.

Se incluye en la carpeta también un makefile para facilitar la evaluación.