Universidad de Puerto Rico
Departamento de Matemáticas
Humacao, Puerto Rico 00791

MATE 4061 | Análisis Numérico | Prof. Pablo Negrón

Laboratorio 2: Evaluación de Polinomios y sus Derivadas

En clase se mencionaron tres métodos para evaluar un polinomio. Vamos a examinar estos métodos en mas detalles partiendo de sus implementaciones en MATLAB. Recuerde que suponemos que el polinomio esta dado por $p(x) = a_1 + a_2 x + \ldots + a_n x^{n-1}$.

Método 1: En este método se calculan primero los términos $a_i x^{i-1}$ y luego estos se suman. Veamos un programa en MATLAB que hace esto:

```
function pval=metodo1(a,z)
n=length(a);
pval=a(1)*ones(size(z));
for k=2:n
    pval=pval+a(k)*z.^(k-1);
end
```

En la inicialización de `pval`, multiplicamos por `ones(size(z))` para el caso en que `z` es un vector de valores en los cuales hay que evaluar el polinomio. Examinando el ciclo "for" en este programa, es fácil ver que el total de multiplicaciones es:

$$1 + 2 + \cdots + (n-1) = \frac{n(n-1)}{2}.$$

Este conteo es para cada elemento de `z`.

Método 2: En el segundo método calculamos x^2, $x^3 = (x^2)x$, $x^4 = (x^3)x$, etc., luego multiplicamos por los coeficientes y sumamos. Esto lo implementamos en MATLAB como sigue:

```
function pval=metodo2(a,z)
n=length(a);
pval=a(1)*ones(size(z));
w=z;
for k=2:n
    pval=pval+a(k)*w;
    w=w.*z;
end
```

Note que en cada iteración del "for" hacemos dos multiplicaciones, lo que nos un total de $2(n-1)$ multiplicaciones.

Método 3: El tercer método, es el método de Horner el cual evalúa el polinomio en su forma anidada. Por ejemplo, el polinomio $p(x) = 2 - 3x + 4x^2 - 5x^3 + 7x^4 + 2x^5$ lo escribimos como:

$$p(x) = 2 + x(-3 + x(4 + x(-5 + x(7 + 2x)))),$$

el cual se evalúa ahora de "adentro" hacia "afuera". En la clase vimos que este método se puede implementar con el programa:

```
function pval=hornerV(a,z)
n=length(a);
pval=a(n)*ones(size(z));
for k=n-1:-1:1
   pval=a(k)+z.*pval;
end
```

En cada iteración del ciclo se hace una multiplicación lo que da un total de n multiplicaciones para el método.

Vamos ahora a comparar estos tres métodos en un caso particular, esto es para la evaluación del polinomio

$$p(x) = 2 - 3x + 4x^2 - 5x^3 + 7x^4 + 2x^5, \tag{1}$$

para $x = -3$. Para cada método, usamos la instrucción `flops` para contar el número de operaciones de punto flotante efectuadas por el método. Trabajamos como sigue:

```
a=[2 -3 4 -5 7 2];
z=-3;
flops(0)
y=metodo1(a,z);
f1=flops
pause
flops(0)
y=metodo2(a,z);
f2=flops
pause
flops(0)
y=hornerV(a,z);
fh=flops
```

Ejercicio 1. *¿Están de acuerdo estos resultados con los conteos operacionales de arriba? Compare estos resultados con los de la función* `polyval` *de MATLAB.* ***Nota****: Escribiendo* `a1=a(6:-1:1)` *obtiene el vector de coeficientes de p en el orden reverso.*

Vamos ahora a evaluar el polinomio en 100000 puntos del intervalo $[-2, 2]$. (Dependiendo de cuan rápido sea el computador que utilice, podría necesitar evaluar en mas puntos). Luego usamos las instrucciones `tic` y `toc` de MATLAB para medir los tiempos de corrida de cada método:

```
z=linspace(-2,2,100000);
tic
y=metodo1(a,z);
t1=toc
pause
tic
y=metodo2(a,z);
t2=toc
pause
tic
y=hornerV(a,z);
th=toc
```

Ejercicio 2. *Calcule* `t1/t2`, `t1/th`, `t2/th`. *¿Qué sugieren estos resultados? Repita este experimento para varios polinomios de grados $n = 5, 6, \ldots, 12$. Haga una gráfica de los tiempos de corrida de cada método versus el grado del polinomio.*

El método de Horner para evaluar un polinomio se puede utilizar para evaluar la derivada del polinomio. Se puede demostrar que si $b_1, b_2, \ldots, b_n$ son generados según el Método de Horner, esto es

1. $b_n = a_n$
2. Para $i = n, n-1, \dots, 1$
   (a) $b_i = a_i + z\, b_{i+1}$,

entonces

$$p'(z) = b_2 + b_3 z + \cdots + b_n z^{n-2}.$$

Ejercicio 3. *Demuestre esto para el caso $n = 3$.*

Modificamos ahora la función `hornerV` para calcular la derivada del polinomio. La función que resulta la llamamos `hornerVp`. Ahora `hornerVp(a,z)` devuelve dos valores: $p(z), p'(z)$. La modificación es como sigue:

```
function [pval,pprime]=hornerVp(a,z)
n=length(a);
pval=a(n)*ones(size(z));
pprime=pval;
for k=n-1:-1:2
   pval=a(k)+z.*pval;
   pprime=pval+z.*pprime;
end
pval=a(1)+z.*pval;
```

En el caso particular de (1), tenemos que $p'(x) = -3 + 8x - 15x^2 + 28x^3 + 10x^4$. Podemos calcular por ejemplo $p'(-3)$ mediante `hornerV([-3 8 -15 28 10],-3)` o como el segundo componente de `hornerVp(a,-3)` donde `a` es el vector de coeficientes de p.

Ejercicio 4. *¿Cúal es el conteo operacional para la función `hornerVp`? Escriba un programa que reciba los coeficientes de p, calcule los de p', y luego use `hornerV` para evaluar p'. ¿Cúal es el conteo operacional para este otro método?*