

EIF400 – Paradigmas de Programación

Proyecto de programación #2

Prof. M.Sc. Georges E. Alfaro S.

PLANTEAMIENTO DEL PROBLEMA

Cuando se están resolviendo polinomios de segundo o tercer grado, se pueden utilizar fórmulas generales para encontrar las raíces de dichos polinomios. Pero cuando se trata de resolver polinomios de mayor grado, es extremadamente difícil hacerlo de esta manera. Un enfoque más práctico utiliza una función que tenga un punto fijo en cada raíz. De esta forma, se puede calcular la raíz alrededor de una primera aproximación de manera recurrente.

FUNCIONALIDAD DEL PROGRAMA

Utilice listas para representar polinomios de grado arbitrario, incluyendo los coeficientes de cada término, comenzado por el término constante.

El polinomio:

$$p_n(x) = \sum_{i=0}^n k_i x^i = k_0 + k_1 x + k_2 x^2 + \dots + k_n x^n$$

Puede representarse con la lista:

$$[k_0, k_1, k_2, \dots, k_n]$$

Por ejemplo: $4 + x^2 - 3x^3 + 2x^5$, se representa como: $[4, 0, 1, -3, 0, 2]$.

Para encontrar las raíces, puede utilizar el **método de Newton-Raphson**, calculando una serie de aproximaciones sucesivas, de la siguiente manera:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

donde x_0 es un valor inicial, próximo a una posible raíz.

Recuerde que una raíz del polinomio $p(x)$ es un valor de x tal que: $p(x) = 0$. No todos los polinomios tienen raíces reales.

Escriba un predicado `calcular/4` que obtenga una aproximación a la raíz de un polinomio arbitrario:

```
calcular(X0, P, E, R).
```

El predicado recibe como parámetros:

X0 : la aproximación inicial de la raíz
P : los coeficientes del polinomio por evaluar
E : error máximo (diferencia máxima entre dos aproximaciones sucesivas)

y regresa:

R : La raíz calculada para el polinomio.

Para poder hacer el cálculo, escriba los predicados para **evaluar** un polinomio en un valor x arbitrario y para poder **derivar** el polinomio. No escriba ningún predicado para calcular potencias, sino que utilice **multiplicación anidada** para evaluar los polinomios.

Por ejemplo, la expresión:

$$p(x) = (x - 2)(x + 4)^2 = -32 + 6x^2 + x^3$$

Tiene dos (2) raíces reales: $x = 2, x = -4$, que se pueden calcular por medio de la sucesión:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{-32 + 6x_n^2 + x_n^3}{12x_n + 3x_n^2}$$

Observe que la expresión en el denominador es la derivada del polinomio original:

$$p'(x) = 12x + 3x^2$$

La consulta:

```
?- calcular(1.5, [-32, 0, 6, 1], 1e-9, X).
```

tiene éxito si la variable X es una raíz del polinomio anterior, tomando 1.5 como valor inicial de la secuencia con una tolerancia de 10^{-9} .

En la implementación de referencia, desarrollada como una posible solución, se obtuvieron los siguientes resultados con diferentes aproximaciones iniciales:

```
?- evaluar(1.5, [-32, 0, 6, 1], D).  
D = -15.125.  
  
?- derivar([-32, 0, 6, 1], D).  
D = [0, 12, 3].  
  
?- calcular(-5, [-32, 0, 6, 1], 1e-9, X).  
X = -4.000000017350174 .  
  
?- calcular(-3, [-32, 0, 6, 1], 1e-9, X).  
X = -4.000000004254663 .  
  
?- calcular(1, [-32, 0, 6, 1], 1e-9, X).  
X = 2.0 .
```

Como otro ejemplo, considere:

```
?- calcular(2, [-2, 0, 1], 1e-9, X).  
X = 1.4142135623730951 .  
  
?- calcular(-1, [-2, 0, 1], 1e-9, X).  
X = -1.4142135623730951 .
```

En este caso, $p(x) = -2 + x^2$.

Como $p(x) = x^2 - 2 = (x - \sqrt{2})^2$, el polinomio tiene dos raíces reales $x = \pm\sqrt{2}$.

Los parámetros $x_0 = -1, x_0 = 2$ son simplemente las aproximaciones iniciales a la raíz. Dependiendo del polinomio particular, es posible que un valor inicial arbitrario no permita calcular ninguna raíz, o que ese polinomio no tenga raíces reales. **Todos los polinomios de grado impar tienen al menos una raíz real.** Defina un valor máximo de iteraciones a calcular para detener la función cuando esta no converja a ningún valor fijo.

OBJETIVOS DEL PROYECTO

El proyecto tiene como objetivo estudiar algunas de las técnicas fundamentales de programación declarativa, especialmente el uso de recursividad y el uso de funciones de punto fijo.

Se pretende también que el estudiante conozca y aplique algunos conceptos básicos del uso de métodos numéricos para la resolución de problemas con técnicas no analíticas.

CONSIDERACIONES DE IMPLEMENTACIÓN

Todos los predicados se escribirán en un único archivo fuente.

Puede utilizar funciones de la biblioteca estándar de SWI Prolog para resolver el problema.

Deberá incluir consultas de prueba para comprobar el resultado de los cálculos con polinomios de grado 2, 3, 4 y 5. Para ello, debe escribir los polinomios y su factorización e incluir el detalle de las pruebas en un documento aparte (MS Word o PDF).

ENTREGA Y EVALUACIÓN

El proyecto debe entregarse **por medio del aula virtual, en el espacio asignado para ello**. La entrega es al finalizar la semana 16 (**viernes 26 de noviembre de 2021**). No se aceptará ningún proyecto después de esa fecha, ni se admitirá la entrega del proyecto por correo electrónico. El proyecto se puede realizar en grupos de **tres personas, como máximo**. Deberán enviar un correo al profesor indicado la lista de participantes del grupo antes del viernes de la semana 14 (12 de noviembre de 2021).

Incluya comentarios en el código de los programas y describa detalladamente cada uno de los predicados definidos. Todos los predicados deberán ser descritos en el código fuente, donde se explique la definición de cada uno y como calcula el resultado

Incluya un comentario al inicio de cada archivo fuente, indicando información básica, como se muestra abajo:

```
%  
% =====  
%  
% (c) 2021  
% version 1.0.0 2021-11-07  
%  
% -----  
% EIF400 Paradigmas de Programación  
% 2do ciclo 2021, grupo 01  
% Proyecto 2  
%  
% 12345678 González Abarca, Adriana  
% 87654321 Montero Rodríguez, Carlos  
%  
% =====  
%
```

En caso de que la aplicación no funcione adecuadamente, efectúe un análisis de los resultados obtenidos, indicando las razones por las cuales el programa no trabaja correctamente, y cuáles son las posibles correcciones que se podrían hacer.

El proyecto se evaluará de acuerdo con la siguiente rúbrica:

| Rubro | Valor | | |
|---|------------------------|--|--|
| Evaluación de los polinomios | 20% | -5% Errores de evaluación (< 2) | -15% Errores de evaluación (> 2) |
| Derivación de los polinomios | 20% | -10% Errores en casos de derivación (< 2) | |
| Implementación del método de Newton-Raphson | 20% | -5% Errores en la implementación de la función de aproximación (< 2) | -15% Función de aproximación mal implementada o errores de validación |
| Evaluación de la condición de finalización por convergencia | 10% | -5% Errores de evaluación (< 2) | |
| Finalización de la evaluación cuando no exista convergencia | 10% | -5% Errores de evaluación (< 2) | |
| Ejecución de pruebas | 20% | (se rebajarán puntos proporcionalmente si las pruebas están incompletas o incorrectas) | |
| Cálculo de raíces complejas del polinomio. | 15% OPCIONAL | | |

OBSERVACIONES GENERALES:

- Los proyectos deben entregarse utilizando el mecanismo indicado por el profesor (el aula virtual, en este caso).
- Se debe indicar en cada documento el **nombre completo y cédula de cada participante del grupo**, indicando el nombre del curso, ciclo lectivo y descripción del trabajo que se entrega, en cada archivo fuente entregado.
- Cualquier trabajo práctico que no sea de elaboración original de los estudiantes (plagio) se calificará con nota 0 (cero) y se procederá como lo indiquen los reglamentos vigentes de la universidad.

REFERENCIAS

Pueden consultar las siguientes referencias:

Nested Evaluation of a Polynomial. (8 de november de 2021). Obtenido de Western Sydney University - School of Computing, Engineering & Mathematics: https://staff.cdms.westernsydney.edu.au/cgi-bin/cgiwrap/zhuhan/dmath/dm_readall.cgi?page=8&part=1

Newton's Method. (6 de noviembre de 2021). Obtenido de Wikipedia: https://en.wikipedia.org/wiki/Newton%27s_method

Polynomial Functions. (8 de noviembre de 2021). Obtenido de Open Calculus: <https://math.dartmouth.edu/opencalc2/cole/lecture7.pdf>