

Métodos Numéricos (2001852)

I Semestre 2016

Laboratorio # 1

Profesor: *Camilo Cubides*

Nombre: _____ Código: _____

Nombre: _____ Código: _____ Calificación: $\frac{\quad}{5}$

Nota: para las funciones que se solicita se programen a continuación en **SciLab**, se debe utilizar, si se necesitan, el método de evaluación de Horner y del cálculo de la derivada de un polinomio, programados en el *laboratorio # 0*.

1. Implementar en **SciLab** una función que permita calcular una raíz real por el método de *bisección*, de un polinomio $p(x)$ cuyos coeficientes estén especificados en un vector **A** para el cual la primera componente define el coeficiente principal del polinomio y el último el coeficiente constante del polinomio, **a** es el extremo izquierdo y **b** es el extremo derecho del intervalo cerrado donde se desea hallar la raíz, **eps** es la precisión deseada y **M** es el número máximo de iteraciones. Para la detención del algoritmo, se debe calcular el error absoluto iterativo del valor funcional $|p(r) - 0| = |p(r)|$ de la aproximación parcial r y asegurar que sea menor que **eps** ($|p(r)| < \text{eps}$) o que se supere el número máximo de iteraciones. La función de **SciLab** debe tener el siguiente encabezado:

`function r = UN_biseccion(A, a, b, eps, M)`

La función debe imprimir una tabla de los cálculos parciales, donde aparezca el número de la iteración, α el extremo izquierdo del intervalo, β el extremo derecho del intervalo, el valor $r = \frac{\alpha + \beta}{2}$ la aproximación parcial de la raíz y el error relativo iterativo. Al terminar de ejecutar la función, se debe informar si se tuvo *éxito* o *fracaso* durante la evaluación y cuál fue la mejor aproximación encontrada.

2. Implementar en **SciLab** una función que permita calcular una raíz real por el método de la *regla falsa*, de un polinomio $p(x)$ cuyos coeficientes estén especificados en un vector **A** para el cual la primera componente define el coeficiente principal del polinomio y el último el coeficiente constante del polinomio, **a** es el extremo izquierdo y **b** es el extremo derecho del intervalo cerrado donde se desea hallar la raíz, **eps** es la precisión deseada y **M** es el número máximo de iteraciones. Para la detención del algoritmo, se debe calcular el error absoluto iterativo del valor funcional $|p(r) - 0| = |p(r)|$ de la aproximación parcial r y asegurar que sea menor que **eps** ($|p(r)| < \text{eps}$)

o que se supere el número máximo de iteraciones. La función de **SciLab** debe tener el siguiente encabezado:

```
function r = UN_regla_falsa(A, a, b, eps, M)
```

La función debe imprimir una tabla de los cálculos parciales, donde aparezca el número de la iteración, α el extremo izquierdo del intervalo, β el extremo derecho del intervalo, el valor r de la aproximación parcial de la raíz y el error relativo iterativo. Al terminar de ejecutar la función, se debe informar si se tuvo *éxito* o *fracaso* durante la evaluación y cuál fue la mejor aproximación encontrada.

3. Implementar en **SciLab** una función que permita calcular una raíz real por el método de *Newton-Raphson*, de un polinomio $p(x)$ cuyos coeficientes estén especificados en un vector **A** para el cual la primera componente define el coeficiente principal del polinomio y el último el coeficiente constante del polinomio, **x0** es el valor inicial, **eps** es la precisión deseada y **M** es el número máximo de iteraciones. Para la detención del algoritmo, se debe calcular el error absoluto iterativo del valor funcional $|p(r) - 0| = |p(r)|$ de la aproximación parcial r y asegurar que sea menor que **eps** ($|p(r)| < \text{eps}$) o que se supere el número máximo de iteraciones. La función de **SciLab** debe tener el siguiente encabezado:

```
function r = UN_newton(A, x0, eps, M)
```

La función debe imprimir una tabla de los cálculos parciales, donde aparezca el número de la iteración, el valor x_i de la aproximación parcial de la raíz y el error relativo iterativo. Al terminar de ejecutar la función, se debe informar si se tuvo *éxito* o *fracaso* durante la evaluación y cuál fue la mejor aproximación encontrada.

4. Implementar en **SciLab** una función que permita calcular una raíz real por el método de la *secante*, de un polinomio $p(x)$ cuyos coeficientes estén especificados en un vector **A** para el cual la primera componente define el coeficiente principal del polinomio y el último el coeficiente constante del polinomio, **x0** es un valor inicial, **x1** es el otro valor inicial, **eps** es la precisión deseada y **M** es el número máximo de iteraciones. Para la detención del algoritmo, se debe calcular el error absoluto iterativo del valor funcional $|p(r) - 0| = |p(r)|$ de la aproximación parcial r y asegurar que sea menor que **eps** ($|p(r)| < \text{eps}$) o que se supere el número máximo de iteraciones. La función de **SciLab** debe tener el siguiente encabezado:

```
function r = UN_secante(A, x0, x1, eps, M)
```

La función debe imprimir una tabla de los cálculos parciales, donde aparezca el número de la iteración, x_i el valor i -ésimo de las aproximaciones de la raíz, x_{i+1} el valor i -ésimo + 1 de las aproximaciones de la raíz y el error relativo iterativo. Al terminar de ejecutar la función, se debe informar si se tuvo *éxito* o *fracaso* durante la evaluación y cuál fue la mejor aproximación encontrada.