Universidad de Concepción
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática

Cálculo Numérico (521230)
Laboratorio 9: Sistemas de ecuaciones no lineales

Ejercicio 1: Consideremos el circuito en la figura 1. Supongamos que

- el voltaje $v_F(t)$ de la fuente de voltage es constante, es decir, $v_F(t) \equiv F$, $\forall t$,
- la corriente $j_R(t)$ y la caída de voltaje $v_R(t)$ a través de la resistencia cumplen
$$j_R(t) = \frac{1}{R} v_R(t) \qquad \forall t,$$
con $R \in \mathbb{R}$ constante,
- la siguiente ecuación describe la relación entre la corriente $j_D(t)$ y la caída de voltaje $v_D(t)$ a través del diodo

$$j_D(t) = \alpha \left(e^{\frac{v_D}{\beta}} - 1 \right) - \mu v_D \left(v_D - \gamma \right), \tag{1}$$

donde $\alpha, \beta, \mu, \gamma \in \mathbb{R}$ son constantes.

Entonces, debido a la topología del circuito y usando las leyes de Kirchhoff se llega a la siguiente relación entre v_D, R y F

$$-\frac{F}{R} + \frac{v_D}{R} + \alpha \left(e^{\frac{v_D}{\beta}} - 1 \right) - \mu v_D \left(v_D - \gamma \right) = 0. \tag{2}$$

Supongamos que conocemos los valores de F y R, la caída de voltaje a través del diodo es entonces el valor de v_D para el cual $F_D(v_D) = 0$ si

$$F_D(v_D) = -\frac{F}{R} + \frac{v_D}{R} + \alpha \left(e^{\frac{v_D}{\beta}} - 1 \right) - \mu v_D \left(v_D - \gamma \right).$$

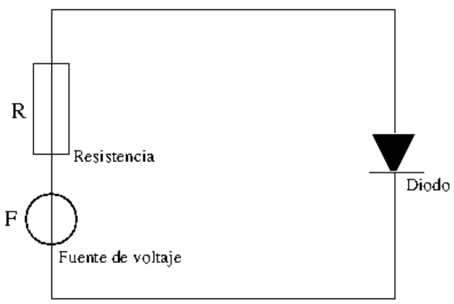

FIGURA 1. Circuito formado por resistencia, fuente de voltaje y diodo.

1.1 Escriba una función MATLAB que, dado $v_D \in \mathbb{R}$ (parámetro de entrada a la función), calcule $F_D(v_D)$. Tome $F = 2$, $R = 10^4$, $\alpha = 10^{-12}$, $\frac{1}{\beta} = 40$, $\mu = 10^{-3}$ y $\gamma = 0{,}8$. Esta función debe estar escrita de modo que, en caso que el parámetro de entrada sea un vector, ella retorne un vector con los valores de F_D en cada uno de los puntos especificados en el vector de entrada.

1.2 Escriba un rutero MATLAB que haga lo siguiente.

a) Evalúe F_D en 100 puntos distintos del intervalo $[0, 0{,}5]$. Haga un gráfico de la función y observe que ella tiene un cero en el intervalo considerado y además $F_D(0)F_D(0{,}5) < 0$.

b) Llame a la función MATLAB `fzero` para obtener una aproximación al cero de F_D en $[0, 0{,}5]$. Ello lo logra escribiendo **una** de las siguientes líneas en su programa, tenga presente que debe sustituir `FD` por el nombre de la función escrita por usted en 1.1.

```
vdfz1 = fzero('FD',0,optimset('TolX',1e-10))

vdfz2 = fzero('FD',[0,0.5],optimset('TolX',1e-10))
```

Para entender los llamados anteriores a `fzero`, lea la siguiente explicación acerca de este comando: `fzero` es una función MATLAB para la aproximación de raíces de funciones reales basada en una combinación del método de bisección con el método de la secante y otras técnicas. *El primer parámetro de entrada* a `fzero` es la función cuya raíz quiere aproximarse. *El segundo parámetro de entrada* puede ser un número real o un vector de 2 componentes. Si es un número real, es considerado como una aproximación inicial al cero de la función. Si es un vector de 2 componentes, representa el intervalo inicial donde buscar el cero de la función (debe cumplir con las exigencias del método de bisección). *Los siguientes parámetros de entrada* son opcionales. En este caso, especificamos que la precisión con que quiere calcularse la aproximación al cero de F_D debe ser 10^{-10} (`'TolX',1e-10`). La forma en que procede `fzero` en cada caso puede leerla escribiendo `help fzero` en la ventana de comandos de MATLAB .
Escriba el valor del voltaje a través del diodo.

v_D	

1.3 Baje el archivo `biseccion.m` de la página de documentación del curso. Llame a esta función para calcular una aproximación al cero de F_D en [0, 0,5] con tolerancia igual a 10^{-10}. Escriba `help biseccion` para entender cómo llamar a esta función y qué valores retorna. ¿Cuántas iteraciones fueron necesarias para alcanzar la exactitud requerida? ¿Cuál es el valor del voltaje a través del diodo?

número de iteraciones (bisección)	
v_D	

1.4 Escriba la función `newton1d.m` que, dados los siguientes parámetros de entrada:
- función MATLAB para evaluar función $f : \mathbb{R} \longrightarrow \mathbb{R}$ cuyo cero desea aproximarse,
- función MATLAB para evaluar $f'(x)$,
- aproximación inicial $x^{(0)}$ a cero de f,
- $tol \in \mathbb{R}$,
- $MaxIt \in \mathbb{R}$,

calcule una aproximación a un cero de $f(x)$ usando $x^{(0)}$ como aproximación inicial. El método debe parar cuando $|x^{(k)} - x^{(k-1)}| \leq tol$ o se hayan realizado $MaxIt$ iteraciones. Recuerde que este método no converge para cualquier escogencia de la aproximación inicial $x^{(0)}$ y es, por tanto, importante, restringir el número máximo de iteraciones a realizar por el mismo. Los parámetros de salida del método deben ser dos: `sol` y `numit`. Ellos deben ser tales que:
- si el método alcanzó la precisión requerida en menos de $MaxIt$ iteraciones, `sol` debe ser la aproximación a un cero de f obtenida y `numit`, el número de iteraciones necesarias para ello.
- Si el método no alcanzó la precisión requerida, `sol` debe ser la última aproximación calculada y `numit` debe ser igual a `-1`.

1.5 Escriba una función MATLAB para evaluar $F'(v_D)$.

1.6 Llame a la función `newton1d` para obtener una aproximación al cero de F en [0, 0,5]. Use $x^{(0)} = 0$ como aproximación inicial, exija una precisión de 10^{-10} y restrinja el máximo número de iteraciones a 20. ¿Logró el método de Newton obtener una aproximación al cero de F con la precisión requerida? ¿Cuántas iteraciones fueron necesarias para ello? ¿Cuál es el valor del voltaje a través del diodo?

número de iteraciones (Newton)	
v_D	

1.7 Llame a `newton1d` con $x^{(0)} \in \{\ 0.2,\ 0.4,\ 0.5,\ 0.8,\ 1\ \}$, fije el número máximo de iteraciones a 20 y observe como cambia el comportamiento del método en dependencia de la escogencia de la aproximación inicial. ¿Para qué valores de $x^{(0)}$ el método no alcanza la precisión requerida en el número máximo de iteraciones dado? ¿Significa esto que el método de Newton no converge para esta escogencia del valor inicial? Cambie el máximo número de iteraciones a 30, vuelva a llamar a `newton1d` con los valores de $x^{(0)}$ para los cuales el método no pudo calcular una aproximación a v_D con la precisión requerida, ¿puede el método ahora encontrar una aproximación a v_D con la precisión requerida?

$x^{(0)}$	número de iteraciones (Newton)
0.2	
0.4	
0.5	
0.8	
1	

Ejercicio 2: Baje de la página de documentación del curso el programa `newtonrd.m`. Observe las diferencias con respecto a la implementación realizada por usted en la pregunta anterior del método de Newton para aproximar ceros de funciones $f : \mathbb{R} \longrightarrow \mathbb{R}$. Utilice este método para aproximar los ceros de la función

$$F(x,y) = \begin{pmatrix} x^2 + xy + y^2 - 1 \\ y - x^2 \end{pmatrix}$$

con una precisión de 10^{-10}. Restrinja el número máximo de iteraciones a 30. Llame a la función `newtonrd` con distintos valores iniciales $(x^{(0)}, y^{(0)})$ y observe qué sucede en cada caso. Complete la siguiente tabla con los resultados obtenidos.

$(x^{(0)}, y^{(0)})$	número de iteraciones (Newton)	aproximación a cero de F

Observación: Note que $F(x,y) = 0 \Leftrightarrow y = x^2 \wedge x^4 + x^3 + x^2 - 1 = 0$. Con ayuda de la función `roots` de MATLAB puede, entonces, encontrar los puntos $(x,y) \in \mathbb{R}^2$ tales que $F(x,y) = 0$ y comparar estas raíces con las aproximaciones obtenidas en los distintos llamados a `newtonrd`.

Ejercicio 3: Resuelva los siguientes problemas

3.1 Encuentre el área entre las curvas $(x, f(x))$ y $(x, g(x))$ con

$$f(x) = e^{x - x^2}, \quad g(x) = \arctan x^2, \quad x \in [-3,\ 3].$$

Observación: Grafique las funciones en $[-3,\ 3]$ y calcule el área entre las curvas utilizando `quad` y `fzero` de manera adecuada.

3.2 Encuentre todos los valores de $x \in [-5,\ 5]$ tales que

$$\int_0^x \sin t^2 \mathrm{d}t = \frac{\sqrt{\pi}}{4}.$$

Utilice para ello `fzero` y `quad` de manera adecuada.