PRÁCTICA DE ALGORÍTMICA NUMÉRICA
PROBLEMAS PUNTO FIJO. MÉTODOS ITERATIVOS Y APLICACIONES

ENTREGA hasta 5/12/18

OBJETIVO: Calcular soluciones de ecuaciones planteadas en forma de punto fijo

$$x=g(x)$$

mediante el método iterativo

$$x_{k+1}=g(x_k)$$

a partir de un x_0 dado.
- Implementar método iterativo.
- Obtener soluciones aproximadas x_k. Si la sucesión de iteraciones $\{x_k\}$ converge a r (lim x_k =r) entonces: r es punto fijo de g(x) (r=g(r)) y la solución buscada.
- Estudiar el error $e_k = |x_k - r| \cong |x_{k+1} - x_k|$ y convergencia.

Parte I. Se va a estudiar la aplicación de varios métodos para calcular la solución de la ecuación no lineal

$$f(x)= (\exp(3x)-1)/x-5=0 \qquad (1)$$

en el intervalo [0.2,1].

Nota: Esta ecuación aparece en un problema planteado en terminología financiera en el documento "Newton-Raphson Method For Solving Nonlinear Equations Part I - Solving Univariate Equations, Gary Schurman, MBE, CFA" en http://www.appliedbusinesseconomics.com/files/gvsnr01.pdf. Textualmente: Our Hypothetical Problem: Start-up company ABC is a typical start-up in that it's revenue growth rate is initially very high (because it's current revenue base and market share are small) but it's fixed costs are such that it is incurring losses. The goal of this company is to grow it's revenue base and become profitable before it runs out of cash. We calculate that the company must take in $50 million in gross revenue over the next three years or else it will fail. Its current annualized revenue base is $10 million. The question becomes " What does the revenue growth rate have to average over the next three years for this company to survive?"

1. - Representar gráficamente la función y=f(x) (en rojo), el eje de abcisas (y=0) (en negro) en el intervalo [0.2,1] y comprobar que f(x) tiene un cero, que llamaremos r (f(r)=0), en dicho intervalo. Atendiendo a la gráfica dar un valor aproximado r.
- Justificar formalmente que existe una raíz de f(x) en el intervalo señalado.

2. **Método 1**. Se escribe la ecuación (1) en forma de ecuación de punto fijo equivalente:

$$x=(\exp(3x)-1)/5=g_1(x)$$

- Aplicar el método iterativo $x_{k+1}=g_1(x_k)$ a partir de x_0=0.4 para calcular la solución iterando 20 veces . El código empleado debe proporcionar en cada iteración: el número de iteración k (%d), la solución x_k obtenida en cada iteración con 16 decimales (%.16f), y una estimación del error e_k cometido en cada iteración (formato: (%.2e). Para estimar el error e_k en la iteración k-ésima se utilizará $e_k = |x_k - r| \cong |x_{k+1} - x_k|$ (Usar comando fprintf con los formatos indicados).

- A la vista de los resultados ¿crees que el método iterativo aplicado converge en este caso?

Nota: La razón es que la función g(x) (regla que implementa el paso de una iteración a otra) empleada en este caso no es *contractiva* en un intervalo que contenga a la raíz. Se dice que g(x) es *contractiva* en un intervalo [a,b] si $|g(x)-g(y)| \le c|x-y|$ con $0<c<1$, $\forall x,y \in [0.2,1]$.

3. **Método 2.** Se escribe la ecuación (1) en forma de ecuación de punto fijo:

$$x=\log(1+5x)/3=g_2(x)$$

- Aplicar el método iterativo $x_{k+1}=g_2(x_k)$ a partir de x_0=0.4 para calcular la solución iterando 20 veces. El código empleado debe proporcionar: el número de iteración k (%d), la solución x_k obtenida en cada iteración con 16 decimales (%.16f), y una estimación del error e_k. cometido en cada iteración (formato: %.2e).

- A la vista de los resultados obtenidos ¿crees que el método iterativo converge en este caso? Comprueba que la función $g_2(x)$ es contrativa en el intervalo [0.2,1]. Para ello, aplicando previamente el Teorema del Valor Medio ($|g_2(x)-g_2(y)| \le |g_2'(\xi)||x-y|$ con x,y,$\xi \in [0.2,1]$), basta comprobar que $|g_2'(x)|$ es estrictamente menor que 1 en [0.2,1].

- Construir la sucesión e_{k+1}/e_k (e(2:end)./e(1:end-1)), redondeando los resultados a un decimal ¿converge a un nº no nulo? ¿De qué orden (velocidad convergencia) es el método?

4. **Método 3.** Aplicar el método de Newton-Raphson $z_{k+1} = z_k - \dfrac{f(z_k)}{f'(z_k)}$ a la función f(x) dada en (1) a partir de z_0=0.4, para calcular la solución iterando 20 veces. El código empleado debe proporcionar: el número de iteración k (%d), la solución z_k obtenida en cada
iteración con 16 decimales (%.16f), y una estimación del error ez_k cometido en cada iteración (formato: %0.2e).
- ¿Crees que el método iterativo converge en este caso? ¿Cuántas iteraciones garantizan que el error es menor que 10^-16?

- Calcular la sucesión de términos ez_{k+1} / $(ez_k)^2$ ¿se aproxima a un nº no nulo? ¿Cuál es el orden de convergencia del método?

- Disponer los resultados obtenidos con los métodos 2 y 3 en la Tabla 1 adjunta.

TABLA 1

Método 2		Método 3	
x_k	e_k	z_k	ez_k

5. Estudio de la precisión y velocidad de convergencia de los métodos 2 y 3. Sea r el valor obtenido de la solución con el método 3. Utilizando comando subplot(1,2,), representar gráficamente en la escala adecuada, los errores relativos (erelxk y erelzk en la misma gráfica, title('Errores relativos')) y el nº de cifras decimales significativas (ncifk y ncifzk en la misma gráfica, title('Cifras decimales significativas')) que producen las soluciones xk y zk obtenidas por los métodos 2 y 3, respectivamente, iterando 80 veces. Comentar los resultados. Cuántas iteraciones son necesarias en cada caso para calcular la solución con una precisión de 15 cifras decimales?

6. - Construir una función *function[x,e,iter]=fun2(x0,tol,nmax)* que implemente la aplicación del método 2 para resolver la ecuación dada, con un error menor que una tolerancia dada *tol,* un nº máximo de iteraciones *nmax* y un valor inicial *x0*. Los parámetros de salida son la solución *x*, una estimación del error que se produce e y el nº de iteraciones calculadas *iter*. Usar bucle while.

- Construir una función *function[z,ez,iterz]=fun3(x0,tol,nmax)* que implemente la aplicación del método 3 para resolver la ecuación dada con un error menor que una tolerancia dada *tol,* un nº máximo de iteraciones *nmax* y un valor inicial *x0*. Los parámetros de salida son la solución *z*, una estimación del error *ez* y el nº de iteraciones calculadas *iterz*. Usar bucle while.

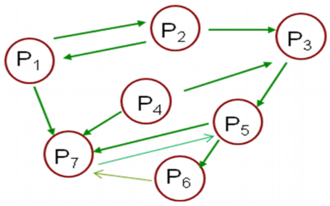
- Aplicar las funciones anteriores para completar los datos pedidos en la Tabla 2. Comentar los resultados.

TABLA 2

nmax=100 tol=e-16	Método 2			Método 3		
	iter	x	e	iterz	z	ez
x0=0.2						
x0=0.4						
x0=0.6						

Parte II. Se va a aplicar el método iterativo de “punto fijo” a una ecuación matricial de la forma Pv=v, siendo P una matriz nxn y v un vector nx1.

Se considera el grafo dirigido que refleja los enlaces de una mini web compuesta de siete páginas web P1, … , P7:



La información de los enlaces del grafo se recoge en la matriz A:

$$A=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \qquad \text{con } a_{ij} = \begin{cases} 1 \text{ si } P_j \rightarrow P_i \\ 0 \quad \text{en otro caso} \end{cases}$$

1. Siguiendo el modelo matemático propuesto por Brin y Page (PageRank de Google), se considera la matriz P=(pij) con

$$p_{ij} = \frac{a_{ij}}{\sum_{k=1}^{7} a_{kj}} = \begin{cases} 1/n_j & \text{si } P_j \rightarrow P_i \quad n_j = n^{o} \text{ páginas a las que apunta } P_j \\ 0 & \text{en otro caso} \end{cases}$$

- Construir (y mostrar) a partir de A la matriz P que resulta de dividir los elementos de cada columna de A por la suma de los elementos de dicha columna (columna k-ésima de P: p(:,k)=a(:,k)/sum(a(:,k)))).

2. Sea v=(v1,v2,v3,v4,v5,v6,v7)’ el vector de pesos, donde vi (vi>0) expresa el peso de relevancia o pagerank de la página P_i, (con condición normalización sum(v)=1). Este vector v es tal que

Pv=v

y se calcula aplicando el método iterativo “de punto fijo” a la matriz P y normalizando en cada iteración (para que la suma de las componentes del vector resultante sea 1, norma 1 de v=norm(v,1)=1):

$$v^0 \text{ inicial (por ejemplo } v^0 = (1/7)*ones(7,1))$$
$$v^{k+1} = Pv^k \qquad \text{\%Iteración}$$
$$v^{k+1} = v^{k+1}/norm(v^{k+1},1) \qquad \text{\%Normalización (sum}(v^{k+1})\text{=1))}$$

- Programar este método para que calcule el vector v con un error menor que e-10 (bucle while) y un nº máximo de 100 iteraciones. Utilizar para estimar el error en cada iteración $e^k = \|v^k - v\| \cong \|v^{k+1} - v^k\|$, donde los vectores anteriores están todos normalizados (la suma de sus componentes es 1 o la norma 1 de dichos vectores es 1).
- ¿Cuántas iteraciones realiza el método?
- Mostrar el vector v obtenido ¿Cuál es la ordenación de las páginas atendiendo a los pesos de relevancia expresados en el vector v, de forma que la página que mayor peso ha obtenido es la 1ª, la que menor peso ha obtenido es la última,…?
- Representar gráficamente, en la escala adecuada, los errores resultantes en las iteraciones realizadas.

3. El vector v buscado es el autovector asociado al autovalor 1 (autovalor dominante de la matriz P). Comprobar que el vector v asociado al autovalor 1 proporcionado por Matlab utilizando el comando eig ([V,E]=eig(P); v=V(:,1)/norm(V(:,1),1)) se parece al vector calculado en el apartado anterior y proporciona el mismo ranking de las páginas web.