

Métodos Numéricos Aplicados a la Ingeniería Civil.

Profesor: Felipe Uribe Castillo.

Taller I: Solución de sistemas de ecuaciones lineales e Interpolación.

Versión No. 1

El plazo para la entrega del taller es hasta el *Miércoles 9 de Marzo (6:00pm)*. Por cada día de retraso en la entrega del trabajo se les descontará 0.3 unidades de la nota final.

Enviar a mi correo, un comprimido que incluya el archivo en word (convertido a pdf) o L^AT_EX con la solución de los puntos y los programas desarrollados. Los programas deberán estar bien comentados, bien identados y demás recomendaciones vistas en clase (se rebajará si el código no cumple con esto). Habrán bonificaciones si ustedes hacen su propia versión de los programas vistos en clase.

1. Con base en las siguientes líneas de código:

```
1 %  
2 x = 1; while 1+x > 1, x = x/2, pause(.02), end  
3 x = 1; while x+x > x, x = 2*x, pause(.02), end  
4 x = 1; while x+x > x, x = x/2, pause(.02), end
```

i). Explique que está haciendo cada línea (ciclo while). *ii).* Cuántas salidas esta produciendo cada ciclo. *iii).* Cuáles son los dos últimos valores de x que se están mostrando en pantalla para cada ciclo.

2. Suponga que 3 personas están conectadas por cuerdas de bungee y se disponen a saltar. Este problema está gobernado por la segunda ley de Newton, donde se puede hacer un balance de fuerzas para cada persona con el fin de encontrar el desplazamiento que alcanzará. Asumiendo que las cuerdas de bungee se comportan de forma lineal, el problema se puede reducir al sistema de 3 ecuaciones lineales:

$$\begin{array}{rcl} (k_1 + k_2)x_1 & -k_2x_2 & = m_1g \\ -k_2x_1 & + (k_2 + k_3)x_2 & -k_3x_3 = m_2g \\ & -k_3x_2 & + k_3x_3 = m_3g \end{array}$$

donde, m_i es la masa de la persona (kg), k_i es la constante del resorte o bungee (N/m), x_i es el desplazamiento de la persona i medido desde la posición de equilibrio (m), y g es la aceleración de la gravedad (9.816 m/s²). Con base en los siguientes datos:

Persona	m [kg]	k [N/m]	posición inicial [m]
Arriba ($i = 1$)	60	50	20
Medio ($i = 2$)	70	100	40
Abajo ($i = 3$)	80	50	60

i). Calcule los desplazamientos x_i , resolviendo este sistema de ecuaciones utilizando los métodos de Gauss, Gauss-Jordan, LU e iteración de Jacobi (solo 4 iteraciones) a mano (no olvide que al final habrá que sumar a la respuesta los valores de la posición inicial, para poder tener el desplazamiento completo). *ii).* Compare sus respuestas utilizando comandos de MATLAB[®] ó los programas vistos en clase.

3. Con base en los siguientes puntos:

$$\mathbf{x}' = \{(-3/2, -14.1014), (-3/4, -0.931596), (0, 0), (3/4, 0.931596), (3/2, 14.1014)\}$$

i). Calcule los polinomios interpoladores de Lagrange y de Newton a mano. ii). Compare con los programas vistos en clase (haga un solo gráfico para ver la aproximación por ambos métodos). iii). Calcule la matrix de Vandermonde del polinomio que pasa por esos puntos a mano (compare con el comando `vander`). Luego de tener la matriz de Vandermonde, resuelva el sistema de ecuaciones (usando descomposición LU - comando `lu` ó programa visto en clase) para hallar los coeficientes del polinomio, y grafique en MATLAB dicho polinomio junto con los puntos (puede usar los comandos `polyfit` y `polyval`). Compare siempre sus respuestas utilizando comandos de MATLAB ó los programas vistos en clase para ver que todo este bien.

4. Haga un programa que implemente la interpolación usando el método de los vecinos más cercanos.

$$y(x) = \begin{cases} y_0 & \text{si } x_0 \leq x < x_0 + \frac{(x_1 - x_0)}{2} \\ y_k & \text{si } x_k - \frac{(x_k - x_{k-1})}{2} \leq x < x_k + \frac{(x_{k+1} - x_k)}{2} \\ y_n & \text{si } x_n - \frac{(x_n - x_{n-1})}{2} \leq x < x_n \end{cases}$$

Use los siguientes puntos para realizar la interpolación:

$$\mathbf{x}' = \{(0, 0), (1, 0.8), (2, 0.9), (3, 0.1), (4, -0.7), (5, 0.85), (6, -0.2)\}.$$

Compare su código con el comando de `yi = interp1(x,y,xi,'nearest')` (haga un solo gráfico para ver que ambos métodos coinciden).

5. Recordemos que los nodos de Chebyshev en el intervalo $[-1, 1]$ están dados por:

$$x'_k = \cos\left(\frac{2N+1-2k}{2N+2}\pi\right) \quad k = 0, \dots, N$$

y en el intervalo $[a, b]$:

$$x_k = \left(\frac{b-a}{2}\right)x'_k + \frac{a+b}{2}$$

Haga una función de MATLAB[®] (`function`) que me regrese los nodos de Chebyshev para cierto número de puntos N y para cierto intervalo $[a, b]$.