Control 1

30/Octubre

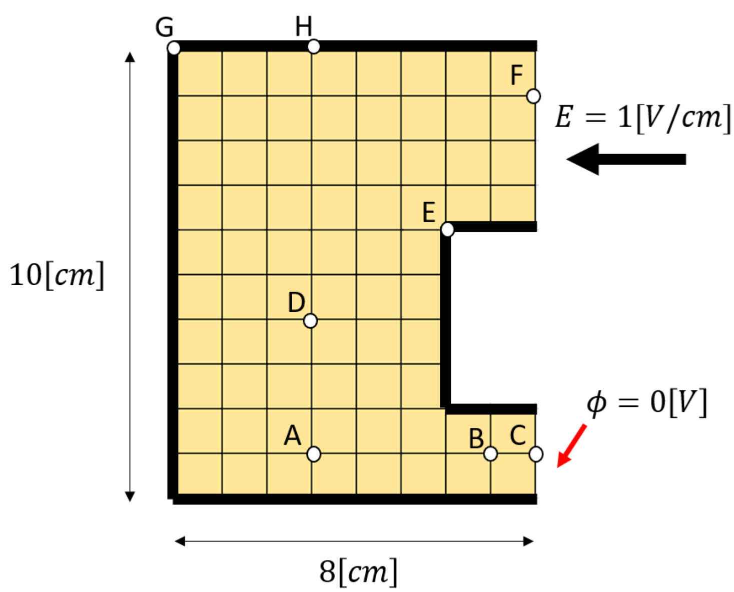
Equipo Docente: Daniel Calderón, Carlos Gonzales, Benjamín Mellado, María José Trujillo, Mauricio Araneda, Pablo Pizarro.

Instrucciones Generales

- Responda cada uno de los 3 problemas en hojas separadas para facilitar la corrección.
- Cada pregunta de desarrollo no debe extenderse más de 5 líneas.
- La última plana incluye un formulario de apoyo.
- Cada problema rebasa en 2.0 puntos de nota. Es decir, si obtiene 2.2, solo recibirá 2.0 puntos.

P1: Métodos Numéricos para EDPs

Se sabe que el potencial eléctrico ϕ cumple con la ecuación de Laplace $\nabla^2\phi = 0$. Se sabe además que el campo eléctrico E cumple con $E = -\nabla\phi$.

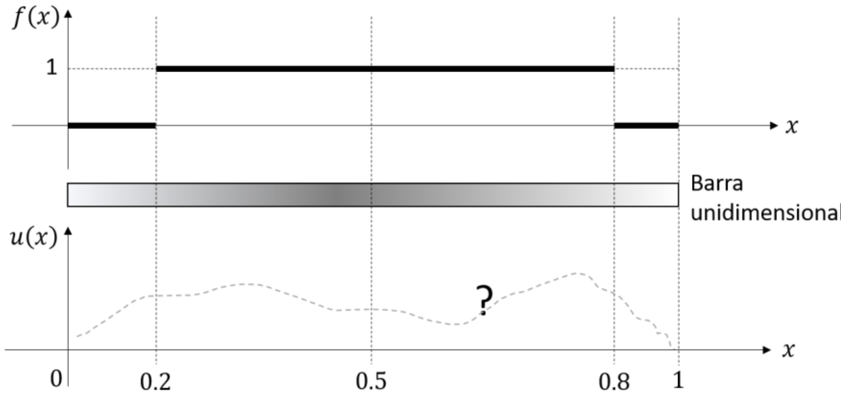


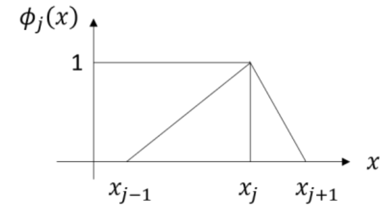
La figura ilustra un corte transversal al interior de un cable perfectamente aislado eléctricamente. En otras palabras, el campo eléctrico es nulo en la dirección normal a los bordes gruesos indicados en la figura.

Resolveremos este problema utilizando el método de diferencias finitas.

a. (0.6) Escriba las ecuaciones de diferencias asociadas a los puntos A, C, D, E, F, G, y H.
b. (0.3) Escriba estas ecuaciones en el sistema matricial $A\phi = b$, descomponiéndolas en, tanto en la matriz A como en el vector b. La numeración de puntos se realiza desde la esquina inferior izquierda hacia la derecha, y luego hacia arriba.
c. (0.2) Si el dominio fuera el doble de alto, pero el mismo ancho. ¿Cuál sería la ecuación asociada al punto B?
d. (0.3) ¿Cómo utilizaría el método de Jacobi para resolver este sistema de ecuaciones? Escriba las ecuaciones de iteración asociadas a los puntos A y B utilizando este método. Invente nombres apropiados para las incógnitas que no conozca.

Ahora consideremos el siguiente problema de conducción de calor en una barra unidimensional, pero utilizando la discretización no regular indicada en la figura.





Sabemos que la temperatura u en la barra cumple con la ecuación:

$$u''(x) = f(x)$$

$$x \in [0,1]$$

$$u(0) = u(1) = 0$$

Hay una fuente de calor f al centro de la barra.

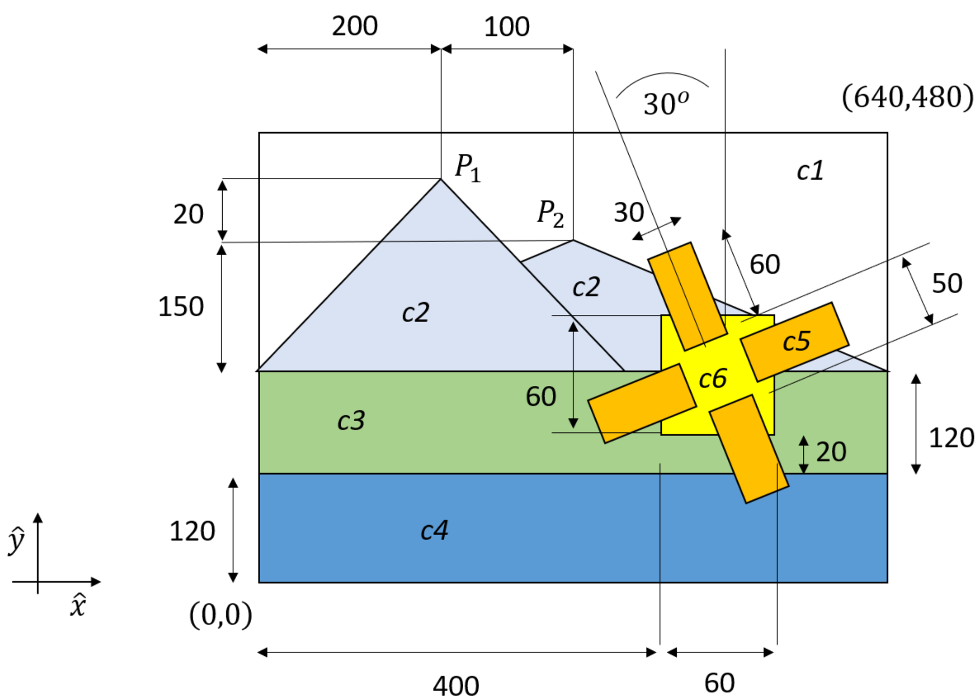
$$f(x) = \begin{cases} 1 & x \in [0.2,0.8] \\ 0 & \sim \end{cases}$$

Utilizaremos el método de elementos finitos con funciones base lineales ϕ_j cómo se vio en clases. Calcularemos $\epsilon_1 = u(0.2)$, $\epsilon_2 = u(0.5)$ y $\epsilon_3 = u(0.8)$.

e. (0.5) Escriba el sistema matricial $A\epsilon = b$ que resuelve el problema. Debe calcular cada componente. Utilice un único valor decimal.
f. (0.1) Bosqueje un gráfico para la temperatura u utilizando las funciones base.
g. (0.2) Justifique 2 ventajas del método de elementos finitos versus el método de diferencias finitas.

P2: Computación Gráfica

a. (0.3) Diferencie entre generación de gráficos real-time versus offline. Proporcione 2 ejemplos para cada caso.
b. (0.2) Describa las 2 tareas ejecutadas por el módulo de Vertex Shading del pipeline gráfico.
c. (0.2) Describa a que corresponden las tareas de Clipping y Screen mapping.
d. (0.3) Un nuevo juego requiere OpenGL 4.6. Revisando la especificación de su computador, nota que dispone solo de OpenGL 4.2. Comente el consejo de su amigo: "Solo debes descargar la última versión de OpenGL. OpenGL es open source, por lo que debe estar disponible para descargar en algún lado."
e. (1.0) Don Pedro quiere visualizar su nuevo molino en acción. Escriba un pseudo código que dibuje la siguiente escena:



Usted dispone de las funciones dibujar:

- dibujarTriangulo(): Dibuja un triángulo isósceles de tamaño 1 de base, y 1 de altura. El origen de este triángulo se encuentra al centro de su base.
- dibujarCuadrado(): Dibuja un cuadrado centrado de tamaño 1x1

Debe utilizar las funciones de transformación de OpenGL. Defina usted los colores c1, c2, …, c6 apropiadamente. Recuerde especificar estos colores antes de dibujar. Las hélices tienen su eje de rotación en el centro del cuadrado que representa el molino.

f. (0.2) Al aplicar la transformación de Shearing $H_{xy}(0.1)$ sobre la escena completa, ¿Cuáles serían los nuevos valores de P1 y P2?

P3: Vertex Blending & Morphing

a. La ecuación de pesos en *Vertex Blending* es la siguiente,

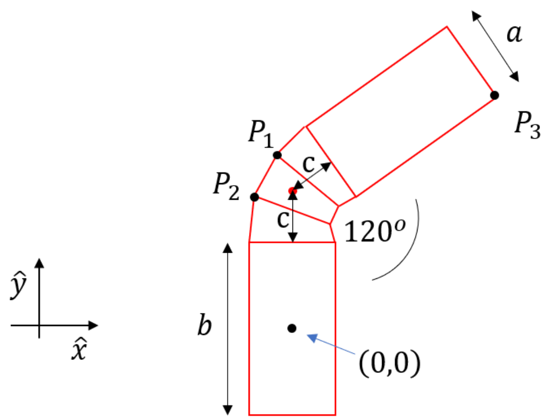
$$u(t) = \sum_{i=0}^{n-1} \omega_i B_i(t) M_i^{-1} p$$

   i. (0.3) Explique cada término de la ecuación.
   ii. (0.4) ¿Qué problema resuelve esta ecuación? Proporcione dos ejemplos de uso

b. (0.3) Describa los dos problemas principales de las estrategias de *morphing.*

c. (0.4) Explique la idea de *morph targets* o *blend shapes*. Presente y describa la ecuación. Bosqueje un diagrama de ejemplo distinto al que se presenta en la lectura.

d. (0.6) Se está modelando un codo con la técnica de *Vertex Blending*, tal como indica la figura.



Ambos rectángulos son iguales y se encuentran separados por una distancia 2c cuando están dispuestos verticalmente. P_1 y P_2 dividen la región intermedia de largo 2c en partes iguales. El rectángulo inferior se encuentra centrado en el origen. P_1 utiliza los pesos 2/3 y 1/3 para las transformaciones asociadas a los rectángulos superior e inferior respectivamente. De manera análoga, P_2 utiliza 2/3 para el rectángulo inferior y 1/3 para el superior.

Utilizando matrices de transformación, deduzca ecuaciones para calcular P_1, P_2 y P_3.

Formulario

$M = N + \sum_{i=1}^{k} \omega_i D_i$	$u(t) = \sum_{i=0}^{n-1} \omega_i B_i(t) M_i^{-1} p$	$m = (1-s)p_0 + sp_1$
$H_{xz}(s) = \begin{bmatrix} 1 & 0 & s \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$p' = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} p$ θ: Antihorario	$u_{i,j-1} + u_{i-1,j} + u_{i+1,j} + u_{i,j-1} - 4u_{i,j} = h^2 f_{i,j}$
$\nabla u = \frac{\partial u}{\partial x}\hat{x} + \frac{\partial u}{\partial y}\hat{y}$	$\nabla^2 u = \frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y}$	$u(\bar{x} \pm h) = u(\bar{x}) \pm hu'(\bar{x}) \mp \frac{1}{2}h^2 u''(\bar{x}) + O(h^3)$
$(v,w) = \int_0^1 v(x)w(x)dx$	$b_i = (f, \phi_i)$	$a_{j,j+1} = -\frac{1}{h_j}$
$u_h(x) = \sum_{i=1}^{M} \epsilon_i \phi_i(x)$	$a_{ij} = (\phi_i', \phi_j')$	$a_{jj} = \frac{1}{h_j} + \frac{1}{h_{j+1}}$

`glPushMatrix()`
 `glPopMatrix()`
 `glColor3f(r,g,b)`
	`glMultMatrixf(…)`
	 `glScale(x,y,z)`
		`glTranslate(x,y,z)`
		 `glRotate(a,x,y,z)`