Examen

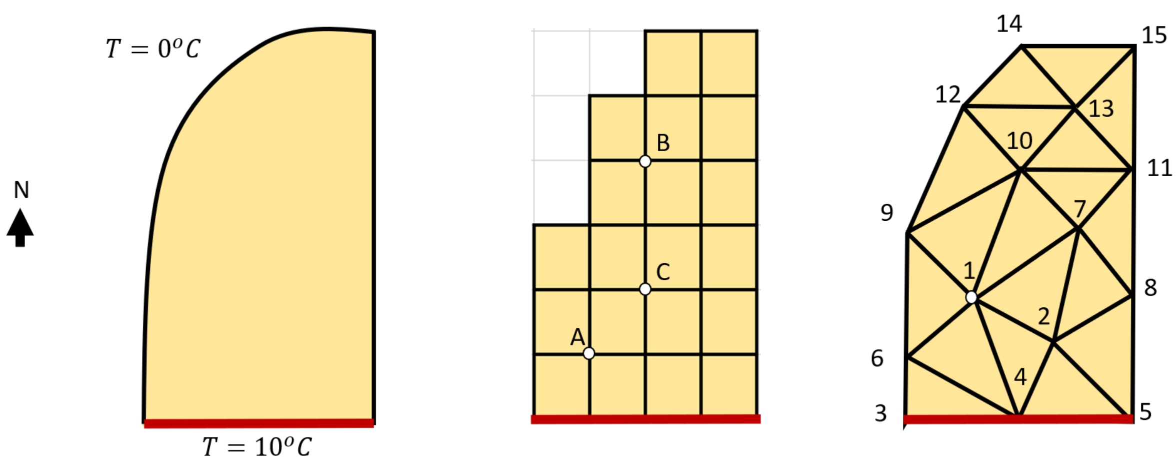
2 de enero, 2019

Instrucciones Generales

- Lea todo el examen antes de comenzar a responder
- La última plana incluye un formulario de apoyo
- Responda concisamente
- Escriba su nombre en cada hoja y numere correctamente cada problema
- El examen incluye 5 décimas extra, su nota final satura en 7
- Tiempo total: 2:30 horas
- El examen tiene 11 problemas

Métodos Numéricos para EDPs

Un refugio en lo alto de una montaña tiene una esquina curva y se encuentra en un ambiente cuya temperatura externa es 0^oC. El muro sur posee una estufa que mantiene la temperatura en 10^oC. Dado que no hay fuentes térmicas en el interior, se sabe que la temperatura T cumple con la ecuación de Laplace $\nabla^2 T = 0$ en el interior del dominio.



Aproximaremos la esquina curva de distintas formas para poder utilizar los métodos de diferencias finitas y de elementos finitos según la figura. Para el caso de diferencias finitas, **la numeración de puntos incógnitos se realiza desde la esquina inferior izquierda hacia la derecha, y luego hacia arriba**. Cada celda tiene ancho y alto igual a 1 metro. Para elementos finitos, utilice los nombres dados para cada punto.

1) (0.8) Escriba las ecuaciones del método de diferencias finitas asociadas a los puntos A, B y C. Escriba estas ecuaciones en el sistema matricial $A\phi = b$, descomponiéndolas en, tanto en la matriz A como en el vector b.

2) (0.5) Escriba la ecuación del método de elementos finitos asociada al punto 1. NO calcule las integrales, exprese la ecuación en función de los coeficientes $a(\varphi_i, \varphi_j)$ cuando no sean nulos. Debe indicar apropiadamente cada índice. ¿Qué términos son conocidos y cuáles son incógnitos?, ¿Cuál es la principal ventaja de este método para este caso?

Modelación

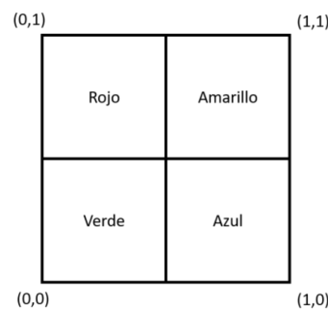
3) (0.4) Compare imágenes raster y vectoriales mencionando dos ventajas y dos desventajas para cada una de ellas

4) (0.4) Describa 2 de los 12 principios de la animación

5) (0.8) Don Pedro quiere construir un suave tobogán que le permita deslizarse hasta el agua del río. Dicho tobogán es modelado utilizando una spline de Catmull-Rom y los siguientes 5 puntos: (0,2), (2,6), (5,2), (10,0) y (13,0). Bosqueje el tobogán. Escriba los polinomios que describen cada uno de los dos trozos del tobogán en función de un parámetro. Escriba también polinomios para los vectores tangentes en cada trozo en función del mismo parámetro.

6) (0.5) En el siguiente archivo OBJ se describe una figura. Bosqueje el modelo representado coloreando (o indicando el color) correctamente según la textura. El primer elemento de cada tipo tiene índice 1.

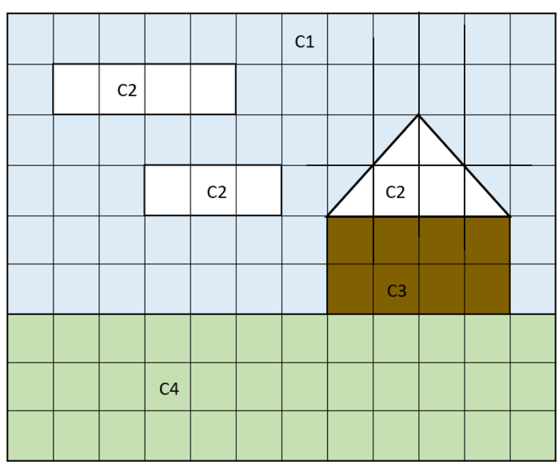
```
v   -10    0      0
v   0      10     0
v   0      0      10
v   0      -10    0
vt  0      0
vt  1      0
vt  0.5    0.5
vt  0      1
vt  0      0.5
f   1/1    4/2    3/3
f   1/5    3/3    2/4
```



7) (0.5) Utilizando OpenGL, escriba un pseudo código que modele un cono como volumen de revolución. Suponga que usted ya se encuentra en el modo GL_TRIANGLES.

Rendering!

8) (0.6) Usted dispone de la función `cuadrado()`, que dibuja un cuadrado de arista 1 centrado en (0,0). Utilizando dicha función, y las funciones de transformación de OpenGL, escriba un pseudo código que modele la siguiente escena:



Suponga que el viewport está configurado con el origen en la esquina inferior izquierda; mide 12 unidades de ancho y 9 unidades de alto. El reticulado, líneas de borde y textos solo tienen fines ilustrativos, NO debe dibujarlos. C1, C2, C3 y C4 indican colores distintos, defínalos usted.

9) (0.6) Los siguientes triángulos T_1 y T_2 ya se encuentran en coordenadas homogéneas de clipping.

$$T_1 \rightarrow (-1,-1,0.5),(1,-1,0.5),(1,1,-0.5)$$

$$T_2 \rightarrow (0,1,0.5),(0,-1,-0.5),(1,-1,-0.5)$$

La ventana de visualización es cuadrada y cada píxel equivale a 0.2 unidades en el espacio de coordenadas homogéneas de clipping, tanto horizontal como verticalmente. Bosqueje una grilla de 10x10 que represente al *frame buffer*. Luego de rasterizar y de aplicar el algoritmo Z-buffer sobre ambos triángulos, anote 1 o 2 en las celdas de la grilla dependiendo del triángulo que se esté visualizando. Deje el fondo sin número.

10) (0.6) Explique brevemente cómo se calcula el color de un vértice al utilizar los modelos de iluminación de Phong, Radiosity y Ray Tracing.

11) (0.8) Explique brevemente cómo se calcula el color de un píxel contenido en el interior de un triángulo para cada uno de los siguientes métodos: Flat, Gouraud, Phong (Sombreado) y Bump-mapping.

$$u(\bar{x} \pm h) = u(\bar{x}) \pm hu'(\bar{x}) + \frac{1}{2}h^2u''(\bar{x}) \pm \frac{1}{6}h^3u'''(\bar{x}) + O(h^4)$$

$$\frac{1}{h_x^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{1}{h_y^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) = f_{i,j}$$

$$B = \frac{\partial u}{\partial x} \approx \frac{u_{i+1,j} - u_{i-1,j}}{2h}$$

$$u_h(x) = \sum_{i=1}^{m} \varepsilon_i \varphi_i(x) \qquad \varepsilon_i = u_h(x_i)$$

$$\sum_{i=1}^{m} \varepsilon_i \, a(\varphi_i, \varphi_j) = (f, \varphi_j) \qquad j = 1, ..., m$$

$$a(u,v) = (f,v) \qquad \forall v \in V$$

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, dx = \int_\Omega \left[\frac{\partial u}{\partial x_1}\frac{\partial v}{\partial x_1} + \frac{\partial u}{\partial x_2}\frac{\partial v}{\partial x_2}\right] dx$$

$$(f,v) = \int_\Omega fv \, dx$$

$$S(\alpha,\beta,\gamma) = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{bmatrix}$$

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad \theta \text{ anti-horario}$$

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$H_{xz}(s) = \begin{bmatrix} 1 & 0 & s \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(T_x,T_y,T_z) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$(M^{-1})^T$$

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

$$Q(t) = \underbrace{\begin{bmatrix} g_1 & g_2 & g_3 & g_4 \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}}_{T(t)}$$

$$\begin{bmatrix} P_1 & P_2 & T_1 & T_2 \end{bmatrix}$$

$$M_H = \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$T_i = \frac{1}{2}(P_{i+1} - P_{i-1})$$

$$C_i(t) = \begin{bmatrix} P_{i-1} & P_i & P_{i+1} & P_{i+2} \end{bmatrix} M_{CR} T(t)$$

$$M_{CR} = \frac{1}{2}\begin{bmatrix} 0 & -1 & 2 & -1 \\ 2 & 0 & -5 & 3 \\ 0 & 1 & 4 & -3 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$M_B = \begin{bmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_1 = P_0 + \frac{T_0}{3}$$
$$P_2 = P_3 - \frac{T_3}{3}$$

$$V - E + F - H = 2(C - G)$$

$$M_{frustum} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & \frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

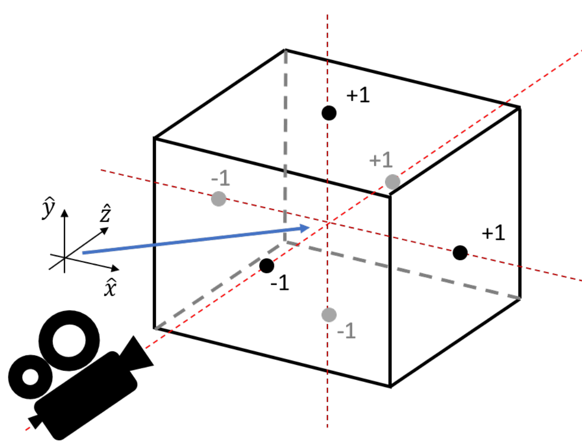
$$M_{Ortografica} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{-2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$I = \frac{1}{a + bq + cq^2}(k_d L_d (l \cdot n) + k_s L_s (r \cdot)^\alpha) + k_a L_a$$

$$n_v = \frac{\sum_{\text{caras vecinas } c} n_c}{\|\sum_{\text{caras vecinas } c} n_c\|}$$

$$L_o(p,v) = L_e(p,v) + \int_{l \in \Omega} f(l,v) L_o(r(p,l), -l)(n \cdot l)^+ \, dl$$

```
f    v/vt/vn    v/vt/vn    v/vt/vn
```



Vertex	
Point	position
FaceRef	face

Face	
VertexRef	vertex[3]
FaceRef	neighbor[3]

Vertex	
Point	position
EdgeRef	edge

Face	
EdgeRef	edge

Edge	
VertexRef	vertex[2]
FaceRef	face[2]
EdgeRef	next[2]
EdgeRef	prev[2]

Vertex	
Point	position
HalfedgeRef	halfedge

Face	
HalfedgeRef	halfedge

Halfedge	
VertexRef	vertex
FaceRef	face
HalfedgeRef	next
HalfedgeRef	prev
HalfedgeRef	opposite