Modelación Computacional - IC0285
Trabajo final

Noviembre de 2011

El trabajo final consiste en tomar el toroide hecho en clase (Figura 1) con los radios r=2 y R=5 y un parámetro de discretización de 5 grados.

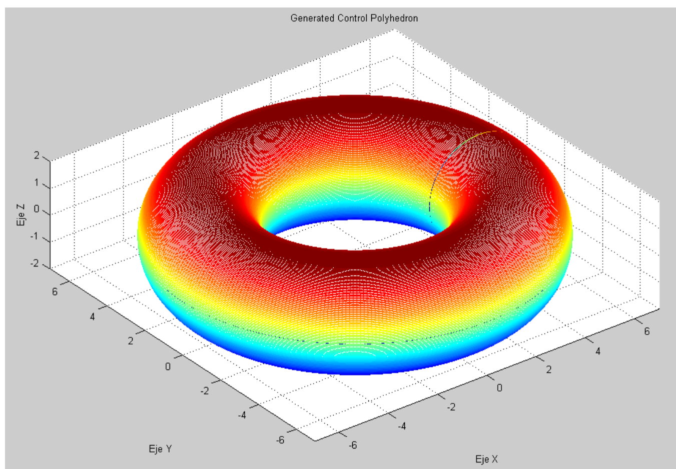

Figura 1: Toroide

La función que crea el toroide, genera 3 matrices X, Y, Z de la forma:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} Y = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1n} \\ y_{21} & y_{22} & \dots & y_{2n} \\ \dots & \dots & \dots & \dots \\ y_{m1} & y_{m2} & \dots & y_{mn} \end{bmatrix} Z = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1n} \\ z_{21} & z_{22} & \dots & z_{2n} \\ \dots & \dots & \dots & \dots \\ z_{m1} & z_{m2} & \dots & z_{mn} \end{bmatrix}$$

1. Primera parte: Transformación mediante matrices de transformación

Partiendo de estas 3 matrices, se pide hacer lo siguiente:

1. Graficar el toroide generado por la función hecha en clase usando el comando 'mesh' de Matlab.

2. Programar la función:

```
function [pts,fils,cols] = empcacar(X,Y,Z)

%entradas: X,Y,Z: matrices entregadas por la función que crea el toroide
%salidas:  pts: matriz empacada que contiene la geometría del toroide
%fils: Número de filas de alguna de las 3 matricesoriginales (X,Y o Z)
%cols: Número de columnas de alguna de las 3 matrices originales (X,Y o Z)
```

   La matriz *pts* tiene 4 filas. Cada fila contiene todos los elementos de las matrices *X,Y,Z*. Esta matriz debe tener la forma:

$$pts = \begin{bmatrix} x_{11} & ... & x_{1n} & x_{21} & ... & x_{2n} & ... & x_{m1} & ... & x_{mn} \\ y_{11} & ... & y_{1n} & y_{21} & ... & y_{2n} & ... & y_{m1} & ... & y_{mn} \\ z_{11} & ... & z_{1n} & z_{21} & ... & z_{2n} & ... & z_{m1} & ... & z_{mn} \\ 1 & ... & 1 & 1 & ... & 1 & ...... & 1 & ... & 1 \end{bmatrix}$$

   Se puede ver que la primera fila contiene todos los elementos de la matriz *X*. En otras palabras, la matriz *pts* tiene la forma:

$$pts = \begin{bmatrix} Fila1_X & Fila2_X & ... & Filam_X \\ Fila1_Y & Fila2_Y & ... & Filam_Y \\ Fila1_Z & Fila2_Z & ... & Filam_Z \\ 1 & 1 & ... & 1 \end{bmatrix}$$

3. Tomar la matriz de transformación M_T :

$$M_T = \begin{bmatrix} 0{,}4249 & -0{,}4223 & -0{,}8007 & 0 \\ 0{,}8823 & 0{,}3911 & 0{,}2619 & 0 \\ 0{,}2026 & -0{,}8177 & 0{,}5388 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Con ayuda de la matriz M_T, se pide hallar un toroide transformado con la siguiente operación:

$$pts_{Trans} = M_T * pts$$

4. Programar la función:

```
function [Xt,Yt,Zt] = desempcacar(pts_trans,fils,cols)
%entradas:  pts_trans: matriz empacada que contiene el toroide transformado
%fils: Número de filas de alguna de las 3 matrices originales (X,Y o Z)
%cols: Número de columnas de alguna de las 3 matrices originales (X,Y o Z)
%salidas: X,Y,Z: matrices desempacadas que contienen la geometría
%del toroide transformado
```

   Las matrixes *Xt,Yt,Zt* deben tener el mismo tamaño de las matrices X, Y, Z (toroide original).

5. Graficar el toroide transformado en una nueva figura (Matrices *Xt,Yt,Zt*) usando el comando 'mesh' de Matlab.

6. Tomar la matriz $Rot = M_T(1:3, 1:3)$ (las primeras 3 filas y las primeras 3 columnas de la matriz M_T). Esta parte de la matriz M_T es la responsable de la rotación del toroide. Hallar los valores y los vectores propios de *Rot* usando el comando 'eig' de Matlab.
   De los 3 valores propios que tiene la matriz *Rot*, hay 2 complejos y un real. Y además de eso, el real vale 1. Ese valor propio real tiene un vector propio asociado; se van a tomar ese valor propio y ese vector propio en particular para definir un ángulo de rotación y un eje de rotación.

7. Programar la función:

```
function [eje_rot,ang] = sel_vector(Vec,Val)
%entradas:  Vec: matriz que contiene los vectores propios de la matriz Rot
%Val: matriz que contiene los valores propios de la matriz Rot
%salidas: eje_rot: eje de rotación del toroide
%ang: ángulo de rotación
```

   La salida eje_rot se refiere al eje de rotación respecto al cual giró le toroide cuando se transformó, es decir, al vector propio asociado al valor propio que vale 1.. Y la salida *ang* se refiere al ángulo que forman la parte real y la parte imaginaria de los otros 2 valores propios (ver comando 'angle' de matlab).
   Entonces, las salidas de la función debe ser el vector propio asociado al único valor propio real y el ángulo de rotación del toroide.

2. Segunda parte: Transformación mediante el método del Cuaternión

El Cuaternion es un método de transformación que toma un vector, y lo gira respecto a un eje de rotación. La figura 2 ilustra el procedimieto.

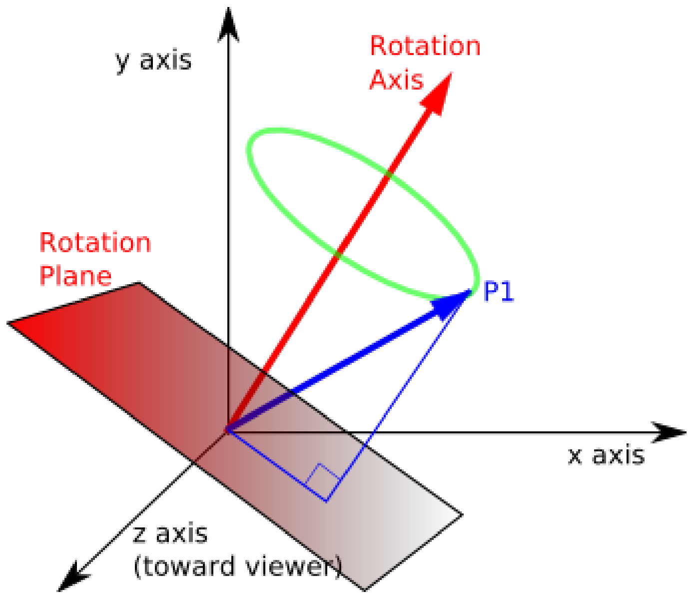

Figura 2: Método del Cuaternión

La ecuación que rige la transformación de un vector P_0, repecto a un eje de rotación eje_rot, se puede expresar así:

$$\bar{P}_f = \bar{P}_0 + 2 * F * (\bar{e} \times \bar{P}_0) + 2 * \bar{e} \times (\bar{e} \times \bar{P}_0) \tag{1}$$

Donde:

$F = cos(\frac{\theta}{2})$
$\bar{e} : eje\_rot * sin(\frac{\theta}{2})$

Nota: el eje de rotación $\bar{e}$ debe ser unitario.

Para realizar la segunda parte, el procedimoento se debe llevar a cabo como sigue.

1. Programar la funcion:

   ```
   function [Bf] = aplicar_cuaternion(pts,eje,ang)
   %entradas:  pts: cuerpo inicial (matriz empacada de 4xn)
   %eje: eje de rotación
   %ang: ángulo de rotación
   %salidas: Bf: cuerpo final (rotado)
   ```

   La función debe entregar un cuerpo transformado B_f (matriz de $4xn$). Para llevar a cabo la transformación basta con aplicar la ecuacion 1 a cada uno de los puntos contenidos en la matriz pts. Siendo las variables eje_rot y θ el eje de rotación y el ángulo de rotación hallados en la sección anterior.

2. Desempacar la matriz B_f usando la función desempacar así:
   $[Xtc, Ytc, Ztc] = desempcacar(Bf, fils, cols)$

3. Graficar el toroide transformado con le método del cuaternión en una nueva figura (Matrices *Xtc,Ytc,Ztc*) usando el comando 'mesh' de Matlab. El toroide transformado con este método debe ser igual al transformado con el método anterior.

3. Consulta sobre transformaciones geométricas

En las 2 secciones anteriores, se usaron transformaciones geométricas. En esta sección, la tarea consiste en buscar una transformación gemométrica y aplicarla.
Las transformaciones que se usen serán matrices de $4x4$ como en la primera sección, la parte mas importante es saber que tipo de transformación se escoge y que implicaciones tiene sobre la geometría final del toroide. La secuencia de trabajo será:

1. Buscar una transformación geométrica en forma de matriz de $4x4$.

2. Programar en Matlab dicha transformación y aplicarla al toroide. En esta parte se tendrá que hacer algo parecido a lo hecho en la primera parte, se tendrá que empacar la geometría inicial del toroide, aplicarle la transformación y luego desempacar la geometría transformada para poder graficarla.

3. Graficar el toroide transformado usando la función 'mesh' de Matlab.

Notas finales:

- Todos los gráficos presentados deben ir en ventanas distintas. Además de eso, todos deben tener nombres de ejes, título y una escala adecuada a la geometría.

- El código de todas las funciones y el archivo principal deben ser debidamente comentados.

- El trabajo se puede hacer en grupos de máximo 4 personas.

- Se debe entregar el día de la entrega:

    - Carpeta comprimida con todos los archivos generados en Matlab.
    - Archivo donde se explique el tipo de transformación que se esogió, que implicaciones tiene sobre la geometría y la fuente de donde se sacó.

- Cada grupo debe trabajar de manera independiente, si 2 grupos entregan el mismo trabajo la notá será de 0.0.

- La entrega se hará el día miércoles 16 de noviembre a las 8:30am en las aulas 23-102 y 23-103.

- La modalidad de entrega será una pequeña exposición por parte de cada grupo, con preguntas para cualquier integrante del grupo. La nota del trabajo se divide en 2 partes: entrega del trabajo 50 % y sustentación 50 %.