Trabajo 1

INFO 265 – Computación Gráfica
Segundo semestre 2015
Universidad Austral de Chile

Objetivo

- Aplicar técnicas de construcción y refinamiento de mallas triangulares.
- Aplicar interpolación lineal para codificar una escala de valores en colores.

Introducción

El trabajo consiste en la construcción de un programa que muestre el gráfico tridimensional de superficies descritas por funciones de la forma $F(x, y)$. Para lograrlo, el programa deberá construir una malla triangular regular sobre el dominio (plano XY), y luego "elevar" cada vértice de la malla según el valor de la función. Un ejemplo de malla (no triangular) así construida se puede observar en la Figura 1.

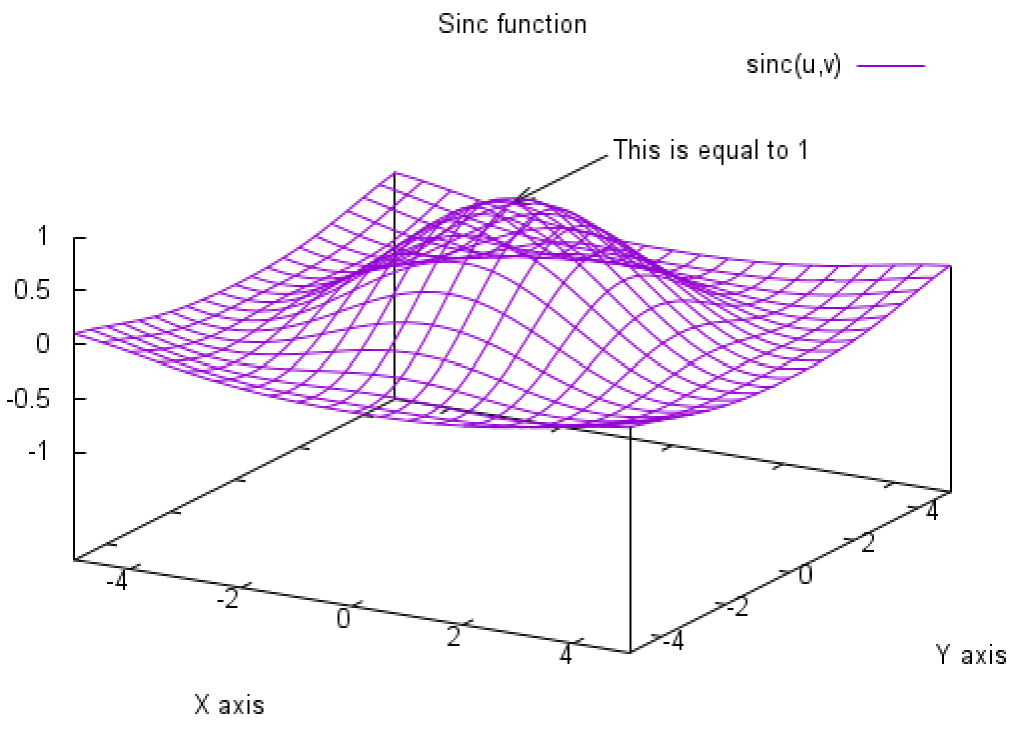

Figura 1: Superficie en gnuplot. Fuente: [1]

Adicionalmente, el programa deberá colorear la malla según la altura de cada triángulo, de modo que se observe el valor de cada punto según una escala de colores. Finalmente, la malla deberá refinarse dinámicamente según la función ingresada, volviéndose más fina para funciones ingresadas que tengan más variación y requieran más triángulos para mantener la fidelidad de la visualización.

Las instrucciones detalladas se enumeran a continuación.

Instrucciones

0 Deberá entregar su trabajo antes del día 16 de octubre (hasta el 16 de octubre a las 8.00 de la mañana). La entrega consiste en dos archivos:

0.1 Archivo *main.js* con su programa.

0.2 Archivo *README* con lo siguiente:

- nombres de los integrantes (máximo 2),
- enlaces a todas las referencias consultadas (indique qué parte fue implementada con ayuda),
- mención de todas las colaboraciones con otros grupos (indique qué parte fue hecha en conjunto), y finalmente
- explicación de cualquier funcionalidad que funcione parcialmente (en caso de que alguna funcionalidad no esté implementada, podrán incluir un comentario breve de cómo podrían resolverlo).

1 (3 ptos) Construya una malla triangular que represente una superficie a partir de la función ingresada: *targetFunction(x, y)*. Para ello, considere que la región de dibujo es el plano XZ en $[-600, 600]^2$ y el eje Y crece hacia arriba (deberá hacer las transformaciones $y \rightarrow z$, $z \rightarrow y$, y alguna transformación del dominio de la función al área de dibujo). Subdivida la región base en triángulos regulares, y eleve cada vértice según los valores de *targetFunction*.

Deberá realizar ajustes sobre el dominio y recorrido de la función *targetFunction*. Para el dominio, seleccione una región arbitraria. Se le recomienda la región $[-5, 5]^2$. Para el recorrido, su ajuste se basará en los valores máximo y mínimo que encuentre de evaluar *targetFunction* en cada vértice.

2 (1 pto) Coloree cada triángulo con un color que represente el valor del promedio de la altura de sus vértices. Construya para ello un procedimiento que reciba un valor y retorne un color saturado, color que varía entre azul saturado para valores mínimos y rojo saturado para valores máximos, pasando por el cyan, verde, y amarillo. Se recomienda usar el espacio de colores HSV, en el que el rojo es el (0,1,1) y el azul el (2/3,1,1) (conversión a RGB implementada en *lib.js*).

3 (2 ptos) Modifique el procedimiento que construye la malla en (1), de modo que la malla construida sea más fina (con más triángulos) si la función *targetFunction* varía más drásticamente. Un esquema de refinamiento simple puede ser el de subdividir cada triángulo en cuatro, obtenidos de unir los puntos medios de los lados (Figura 2).

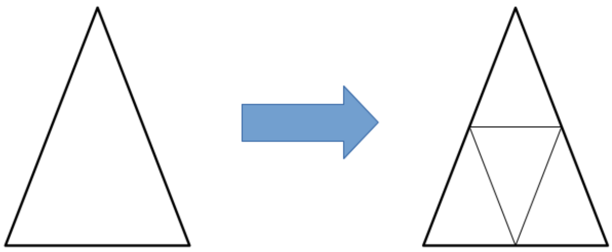

Figura 2: Subdivisión simple de triángulos

El criterio de subdivisión a usar será la condición que se debe cumplir para subdividir la malla. Implemente el siguiente criterio de subdivisión, o algún otro si obtiene con él mejores resultados:

> Si la diferencia entre la función objetivo *targetFunction* aplicada en puntos internos de un triángulo y el valor de la alturas en esos puntos difieren por más que un valor predefinido, se subdividirá la malla. Los puntos a seleccionar pueden ser los puntos medios de los lados del triángulo, y el punto medio de todo el triángulo.

Como prueba inmediata, considere estas dos situaciones ejemplares:

- La función $F(x, y) = x + y$ representa un plano y puede visualizarse con gran fidelidad con sólo dos triángulos, en cambio
- la función de Himmelblau (Figura 3) es una función multimodal y probablemente necesite una malla fina para visualizarse con fidelidad.

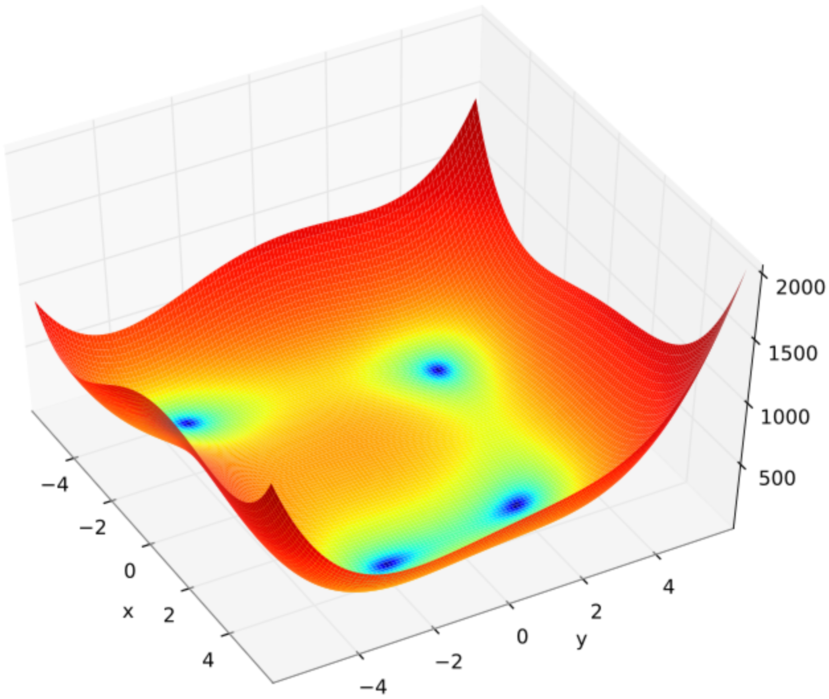

Figura 3: Gráfico de la función de Himmelblau.
Fuente: Wikimedia Commons

Referencias

[1] Gnuplot demos. http://gnuplot.sourceforge.net/demo_5.0/

[2] Test functions for optimization. https://en.wikipedia.org/wiki/Test_functions_for_optimization