

## 1. 目的

本研究の目的は、昨年度の幸谷研究室の松井宏晃先輩が作成した PHPlot を用いたグラフ描画機能付き関数表プログラムの機能拡張である。昨年度のは2次元の線グラフしか描画できなかったものを3次元化し、立体的なグラフ描画を可能にすることを目標としている。

PHPlot は折れ線グラフ・棒グラフ・円グラフなどの1変数関数の2次元グラフの描画が可能であるが、3次元描画ができない。そのため、PHPlot とは異なるライブラリを用いた3次元グラフ描画システムを目指した。

## 2. システム概要

本研究では当初 WebGL・OpenGL・Three.js の3つの3次元描画ライブラリを用いてそれぞれシステムを開発する予定だった。しかし、WebGL は理解が及ばず開発が困難と判断し開発を断念した。以下、OpenGL と Three.js を用いた2つのシステムについて概要を述べる。

プログラムの流れは2つとも同じである。描画する関数を選択し、描画の範囲を設定する。実行すると設定した範囲の座標を算出し、頂点を結びポリゴンを描画していく。

描画モードは OpenGL については1変数関数を Z 方向へ押し出した2次元立体視モードのみ、Three.js は2次元立体視モードと Z 座標の値により選択された関数の係数が変動する3次元モードの2種類を実装した。

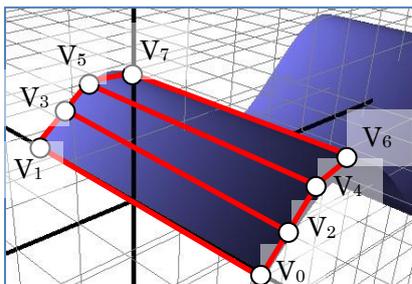
本研究では一変数関数の実装しか頭の中に無かったため、特定の関数しか実装ができなかった。

以下、OpenGL と Three.js を用いた2つのシステムの解説を行う。

### 2.1 OpenGL

OpenGL とは、3次元図形を高速に描画する専用ハードウェアのソフトを開発するにあたり、手間を省くために、簡潔にプログラムできるように設定されたインタフェースであり、2次元と3次元のコンピュータグラフィックスの両方を扱うことができる。日常生活では携帯電話や PDA、家電などに使用されている。

描画方法は指定されて点を結んでゆき、1つのポリゴンを生成するというものである。下図はポリゴンが描画される形を簡単に示したものである。



座標の計算式は以下の通り。x と polygon\_width は、システムの最初に行うデータ入力により設定された値が入る。

$$X = x * \pi$$

$$Y = \text{選択された関数}(X)$$

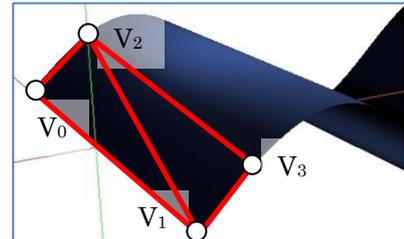
$$Z = \text{polygon\_width} / 2$$

(V1 V3 V5 V7 側は Z 座標が反転した値が入る。)

### 2.2 Three.js

Three.js とは JavaScript のオープンソース・ソフトウェアで、3次元のコンピュータグラフィックスを扱うことができ、WebGL と Canvas を使用できる。

描画方法が OpenGL と異なり、3角形のポリゴンを連続して描画し、形を形成していくというものである。下図はポリゴンが描画される形を簡単に示したものである。



2次元立体視モードの式は同じなので省略する。3次元モードの座標計算式は以下の通り。x と z は入力フォームで入力された値が入る。

$$X = x * \pi$$

$$Y = \cos(Z) * \text{選択された関数}(X)$$

$$Z = z * \pi$$

## 3. ベンチマーク

OpenGL と Three.js で開発した2つのシステムの描画スピードの性能差があるのか確かめるためにベンチマークを行った。実験を行った環境は以下の通り。

OS : Windows7 64bit  
 CPU : IntelCore i5 2.80GHz  
 メモリ : 4.0GB  
 グラフィックボード : NVIDIA GeForce GTX460

### OpenGL

- VisualStudio2010
- glut ver.3.7.6

### Three.js

- Three.js r47
- Firefox ver.9.0.1
- 描画は Canvas を使用

データを入力し終えた後から描画が完了するまでの時間を100回計測する。

計測した結果、それぞれの平均時間は OpenGL が平均 0.568sec、Three.js が平均 6.409sec となった。この結果は、OpenGL が GPU と直接データのやり取りを行うのに対し、Three.js は JavaScript で計算された後で結果を GPU へ送るため、差が生じていると考えられる。また、OpenGL は設定された頂点たちを結んでいき1つのポリゴンを生成するのに対し、Three.js は三角形のポリゴンを連続して生成し、同じ形を模すことで同様の結果を導き出している。

Three.js が描画の手順を多く要しているため時間がかかるということも要因の一つではないかと考えられる。

OpenGL の Web 版ともいえる WebGL を考慮すると、WebGL の開発を行っていたら動作速度は OpenGL < WebGL < Three.js となっていたのではないかと考えられる。

## 4. 今後の課題

本研究では1変数関数のグラフ化しか実装することができなかった。2変数関数を組み込めればさらに描画の種類を増やすことが可能となるだろう。