

## SciQLOP - Plot\_and\_opengl - # 6

### h1. Plot and OpenGL in QtCharts

The following test has been used to define the limits of OpenGL :

```
typedef struct attribute((packed)) dbl_str{
uint64_t mant:52;
uint64_t exp:11;
uint64_t sign:1;
}dbl_str;

typedef union dbl{
double dblval;
uint64_t intval;
dbl_str strval;
}dbl;

QT_CHARTS_USE_NAMESPACE

int main(int argc, char *argv[])
{
QApplication a(argc, argv);

QVector<double> timeVector;
dbl offset;
offset.strval.sign=0;
offset.strval.exp=0b011111111111;
offset.strval.mant=0b1000000000000000000000000000000000000000000000000000;
for(int i = 0; i<(1<<20);i++)
{
    timeVector.append(offset.dblval);
    offset.strval.mant+=1<<(52-24);
}

QLineSeries *seriesOGL = new QLineSeries();
seriesOGL->setUseOpenGL(true);

for(int i=0;i<timeVector.count();i++)
{
    double LUT[]={0.0,1.0,-1.0,2.0,-2.0,3.0};
    seriesOGL->append(timeVector.at(i), LUT[i%6]);
}

Chart *chart = new Chart();
chart->legend()->hide();

chart->addSeries(seriesOGL);

chart->createDefaultAxes();
chart->setTitle("Simple line chart example");

ChartView *chartView = new ChartView(chart);
chartView->setRenderHint(QPainter::Antialiasing);

QMainWindow window;
window.setCentralWidget(chartView);
window.resize(1400, 1300);
window.show();
return a.exec();
}

By changing this line :
offset.strval.mant+=1<<(52-23);
to offset.strval.mant+=1<<(52-24);
```

we observed that the plot did not take into account any changes in the double mantissa after 28 bits (52-24), i.e. some points are stacked because the plot cannot make the difference between two different abscissa values.

This corresponds to the size of the float mantissa.  
We can then assume that the OpenGL plot uses floats.

We found the lines where the doubles are casted to floats in the QtCharts code.  
This takes place in `glxyseriesdata.cpp` in `GLXYSeriesDataManager::setPoints` : each x and y of the points are casted to floats.

The new float vector is then used by `glwidget.cpp` in vertex and fragment source code called by `GLWidget::paintGL`