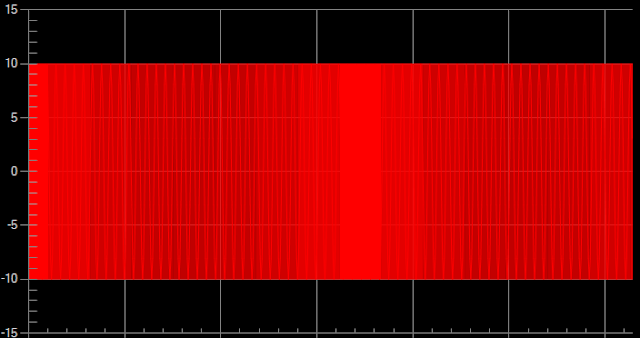
# Implementation of Downsampling Algorithm in MSChart Extension

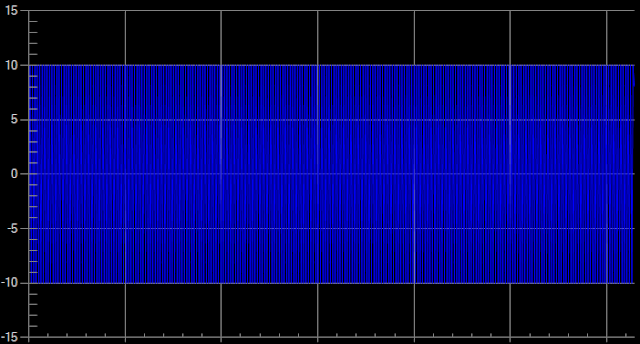
## Introduction

Throughout the years of development is MSChart Extension, performance issue on displaying large data size was still remain unsolved. When refreshing a chart on application, we want it to draw as fast as possible to minimize user waiting time and minimize memory consumption. When working with Microsoft Chart Control (MSChart) in WinForms, the amount of memory used and time to draw a chart gets longer with increased of data size. FastLine series have better performance but still suffer from performance hit when data size growth beyond 100K data points.

Moreover, plotting large data points on screen with limited pixels caused all the data points and lines overlapped, resulting misinterpretation of information to user. Example below shows sinewave with constant frequency of 100K data points compare to downsampled chart with 500 data points.

[](https://1.bp.blogspot.com/-IIdjK9ZUgrY/XyPKFrGFojI/AAAAAAAABkE/0ePwaqHKs28gnxD6PKnj1vmEvk_UFYtugCPcBGAYYCw/s704/SineWave100K.PNG)

**Sinewave with 100K data points**

[](https://1.bp.blogspot.com/-RR3NQ9pjZt0/XyPKLKCy0HI/AAAAAAAABkI/l-BN5C7UZow8Dw7n0ZzP_rYL-aEJscDAgCPcBGAYYCw/s702/SineWaveDownSample500.PNG)

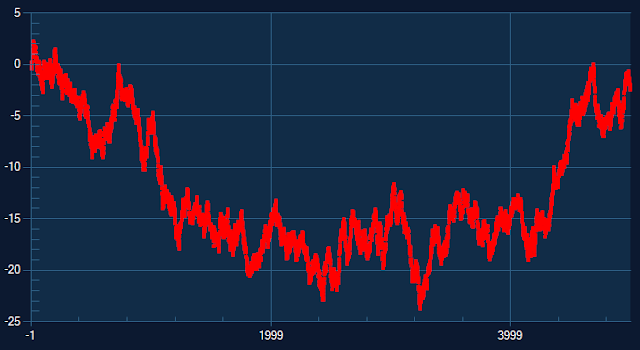
**Sinewave downsampled to 500 data points**

## About Largest-Triangle-Three-Bucket (LTTB) Algorithm

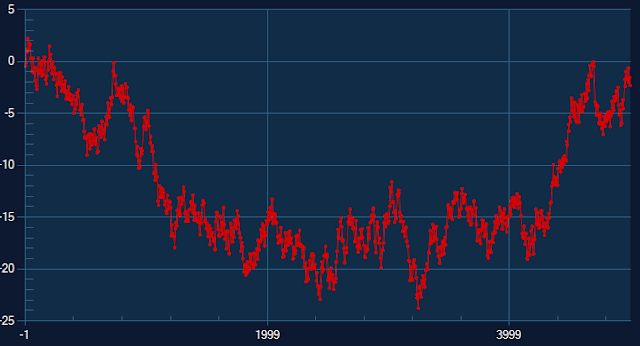
It's clear that we need some sorts of downsampling strategy to plot just enough data points on screen to gain better speed while preserve the peak and trough of the waveforms. It's important to remember that the downsampled data is only meant for data visualization purpose only, not for quantitative analysis.

In MSChart Extension, we decided to implement the [Largest-Triangle-Three-Bucket](https://github.com/sveinn-steinarsson/highcharts-downsample" \t "https://www.codearteng.com/2020/08/_blank) (LTTB) algorithm described by **Sveinn Steinarsson** in his Master Thesis. The complete thesis and implementation on different program language is available in **Sveinn Steinarsson** [GitHub](https://github.com/sveinn-steinarsson/highcharts-downsample" \t "https://www.codearteng.com/2020/08/_blank) page. Implementation in MSChartExtension is based on C# code by [Adrian Seeley](https://gist.github.com/adrianseeley/264417d295ccd006e7fd" \t "https://www.codearteng.com/2020/08/_blank).

The algorithm works with three buckets at a time and proceeds from left to right of divided buckets to select one point with largest effective area to represent each bucket in the downsampled data. Example below shows random data series with 5000 data points and downsampled of the identical series to 500 data points.

[](https://1.bp.blogspot.com/-bBG9ueIpyVY/XyK7HR9LacI/AAAAAAAABjo/7R92-BxbPJUFkKzbbZUrnDx2GR4FoujygCPcBGAYYCw/s728/ChartRandom5000.PNG)

Random data 5000 points

[](https://1.bp.blogspot.com/-9M74ktVf_fY/XyK7HCaPZ5I/AAAAAAAABjk/4fBCQsCb6oMxhADFGKWdkr4K-TQl_fyFQCPcBGAYYCw/s721/ChartRandom800.PNG)

Random data downsampled to 500 points

## DynamicX-Largest-Triangle-Three-Bucket (DLTTB) Algorithm

If you follow **Sveinn Steinarsson's** thesis and GitHub page, you shall be aware that the LTTB have the following limitation.

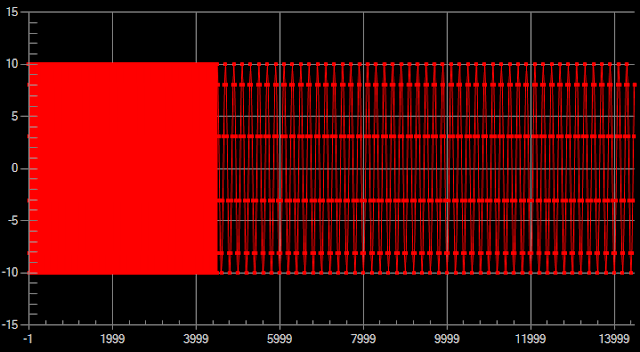
* Does not support gaps (null values) in the data array.
* X-values must be in a strictly increasing order.

We decided to take one step beyond to address the limitation described in the original LTTB algorithm.

Instead of dividing the data points into buckets by data index, the improved **DyanmicX-Largest-Triangle-Three-Bucket (DLTTB)** algorithm divide and group data points into each buckets with fixed X interval. The number of data points in each bucket might vary depends on the data. A bucket might have no data if none of the data point x value fall between the boundary of the bucket.

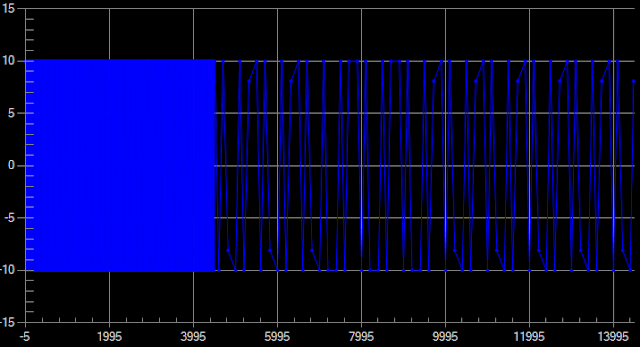
Bucket with zero data point will not be plotted while downsampled data for other buckets will be calculated based on LTTB algorithm. Taking into consideration that some of the bucket might be empty, the downsampled data from DLTTB algorithm might have data points less than the defined display data size.

Example below show 5000 data points of sinewave with 2 different interval where first 4500 data have x increased by 1, while the last 500 data points is having x increased by 20. Look at the amount of data in first 4500 data points which painted it as solid block

[](https://1.bp.blogspot.com/-Q6nlWKH9Nzo/XyPhsCof9aI/AAAAAAAABko/LPF04G9MzH00YLynRc7BkuvP3rnKPAK-QCPcBGAYYCw/s719/ChartOriginalData.PNG)

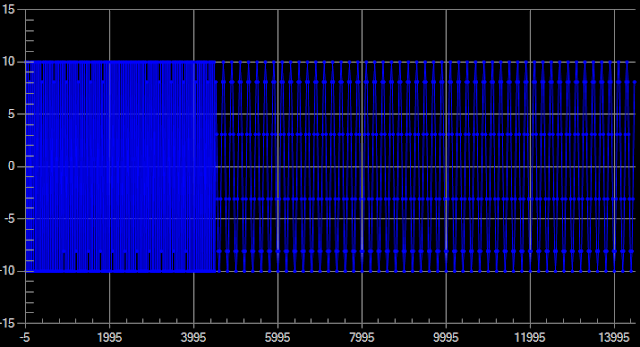
**Original data with 5000 points**

Image below shows downsampled of original data to 800 data point with LTTB algorithm. The last 500 data points looks ugly and loosing important details since LTTB does not aware on different X interval in the data.



**Downsampled 800 points with LTTB**

The next image shows downsampled series to 800 data points with DLTTB algorithm which looks better.



**Downsampled 800 points with DLTTB**

The performance of DLTTB and LTTB is identical as both algorithm used single pass method without involve complex multiplication calculation.

## Implementation In MSChart Extension

LTTB and DLTTB algorithm is implemented in [DownSampling](https://github.com/Code-Artist/MSChartExtension/blob/master/MSChartExtension/DownSampling.cs) class in MSChartExtension. Setting dynamicX between true (DLTTB) and false (LTTB) to select different algorithm. A new option named BufferedMode is introduced in ChartOption class.

To use buffered mode, first enable BufferedMode and set DisplayDataSize in ChartOption when calling method EnableZoomAndPanControls as shown below:

new ChartOption()  
{  
    ...  
    BufferedMode = true,  
    DisplayDataSize = 800  
    ...  
});

Next, add data points to series using the following extension method for Series class.

series.AddXYBuffered( xvalue, yvalue);

With buffered mode, maximum data points to display on chart is defined in DisplayDataSize properties. When user zoomed in the chart, number of visible data on scale view will be re-evaluate. Downsampling algorithm will take effect if number of visible data is more than twice of defined display data size. Otherwise, all visible data will be plotted. With such, details of the data will made available to user when zoomed in. First and last point of the series is always plotted to ensure the PAN function works correctly.

Article published at:

* <https://www.codearteng.com/2020/08/implementation-of-downsampling.html>