gPCD Data Analysis

Best Fit Trendlines and Analysis

Table of Contents

Performance Analysis
Summary of interpretation

The data being analyzed is the time to detect collisions while the number of particles and collisions grow. The collisions are always half of the number of particles so that they grow linearly with the number of particles.

1. It is probably not possible to get an exact data fit for the frame times because the GPU makes decisions based on the size, and/or organization of the data.

2. The data is heteroscedastic so that variation in time grows as the time scale grows. A log plot was performed to remove scale variations. It showed that that the time variation is does grow in direct proportion to the number of particles which is a feature of the GPU mechanics, data size and/or data organization.

Performance Analysis
Summary of interpretation

The data being analyzed is the time to detect collisions while the number of particles and collisions grow. The collisions are always half of the number of particles so that they grow linearly with the number of particles.

1. Total time is the sum of the graphics and compute pipeline processing time. The fit can be impacted by the difference in the observed complexities of each. But the theoretical complexity of total time is known to be $\mathcal{O}(n^2)$.

2. A very small base cost per frame (bcpf) (K) puts a tiny interval very close to the minima of the fit function such that it is essentially linear. The trendlines must be considered “as if and only on this interval” or ‘if there were a competing algorithm with this base cost, and this interval, what would it look like?”.

3. The degree of parallelization can grow to a point where there is an ***effective*** reduction of order from the original complexity (in this case $\mathcal{O}(n^2)$) to a linear one ($\mathcal{O}(n)$ in this case).

Performance Analysis
Summary of interpretation

The data being analyzed is the time to detect collisions while the number of particles and collisions grow. The collisions are always half of the number of particles so that they grow linearly with the number of particles.

1. A competing quadratic algorithm for the frame time of the graphics pipeline, has a base cost of 2.7785E-18(s) where *x* is the number of particles and *t(s)* is the time to process one frame *($t = 2.7785E\text{-}18x^2 + 1.3402E\text{-}09x - 5.4226E\text{-}05$, $R^2 = 9.9191E\text{-}01$).*

2. A plot of the residuals of the quadratic fit functions for the graphics shows a periodic variation around the mean that increases with time but shows no other tendencies.

Performance Analysis
Summary of interpretation

The data being analyzed is the time to detect collisions while the number of particles and collisions grow. The collisions are always half of the number of particles so that they grow linearly with the number of particles.

1. A competing linear algorithm for the frame time of the graphics pipeline, has a base cost of 1.3624E-09(s) where *x* is the number of particles and *t(s)* is the time to process one frame (*t* = 1.3624E-09x - 7.4949E-05, R^2 = 9.9188E-01).

2. A plot of the residuals of the linear fit functions for the graphics shows a periodic variation around the mean that increases with time but shows no other tendencies.

Performance Analysis
Summary of interpretation

The data being analyzed is the time to detect collisions while the number of particles and collisions grow. The collisions are always half of the number of particles so that they grow linearly with the number of particles.

1. Notice the similarity between the second term of the quadratic (*1.3402E-09(s))* and the first term of the linear (1.3624E-09(s) - they are essentially the same on this interval.

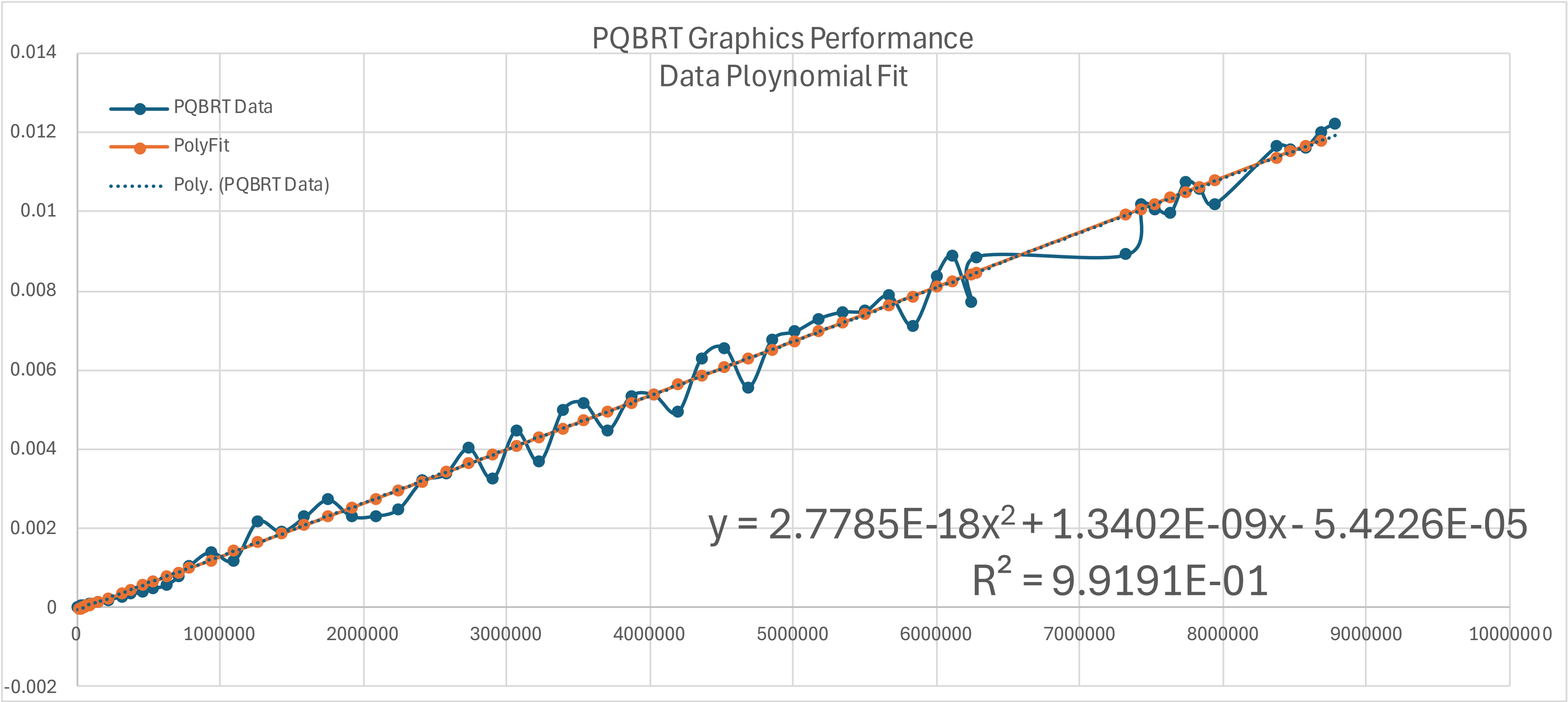
2. A plot of the residuals of the quadratic and linear fit functions are almost identical.

The Graphics Pipeline

Performance data interpretation

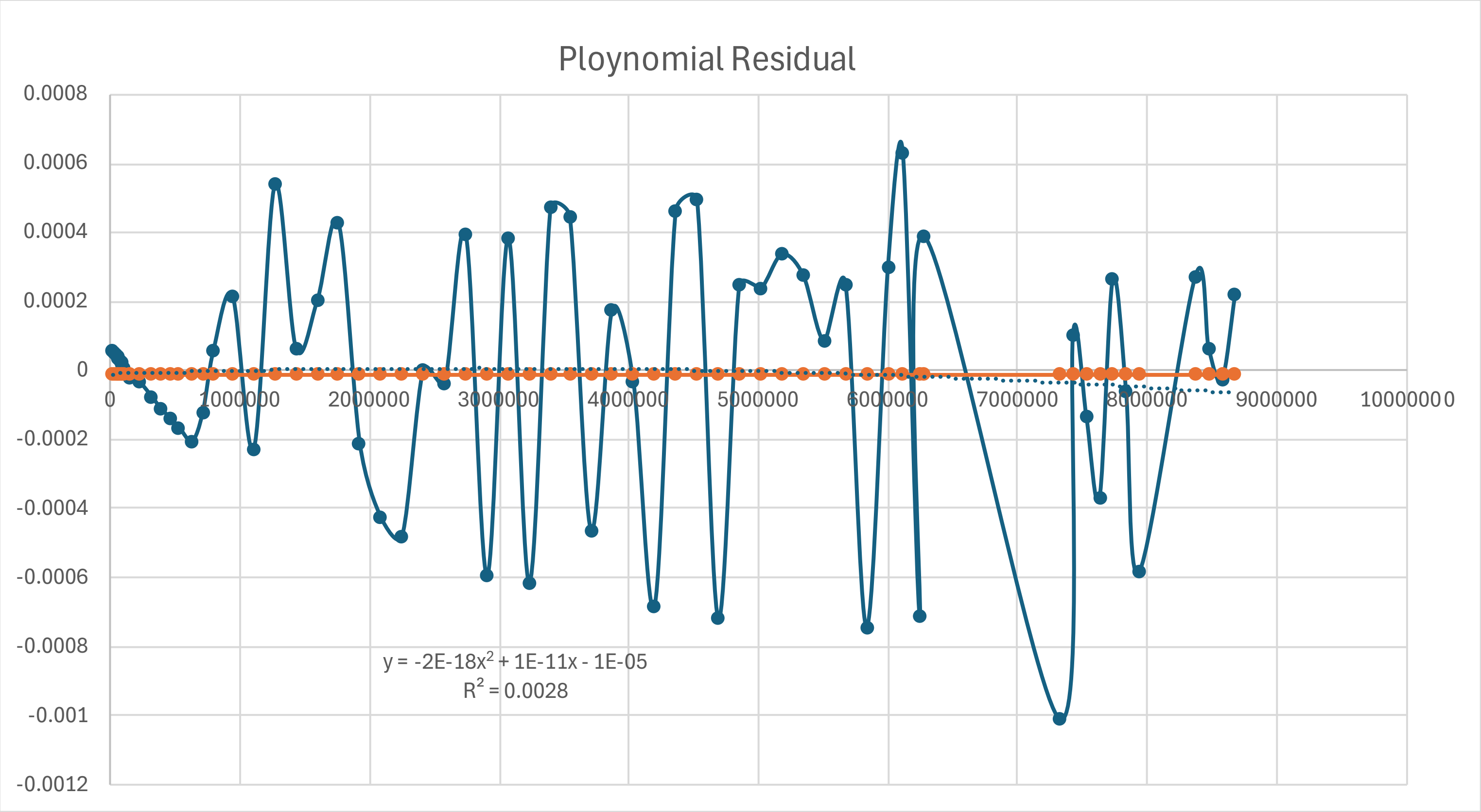
Graphics Pipeline Total Frame Time (spf)
Polynomial Fit

The quadratic trendline for the graphics pipeline. A completing algorithm has a K=2.7785E-18 which is essentially zero.



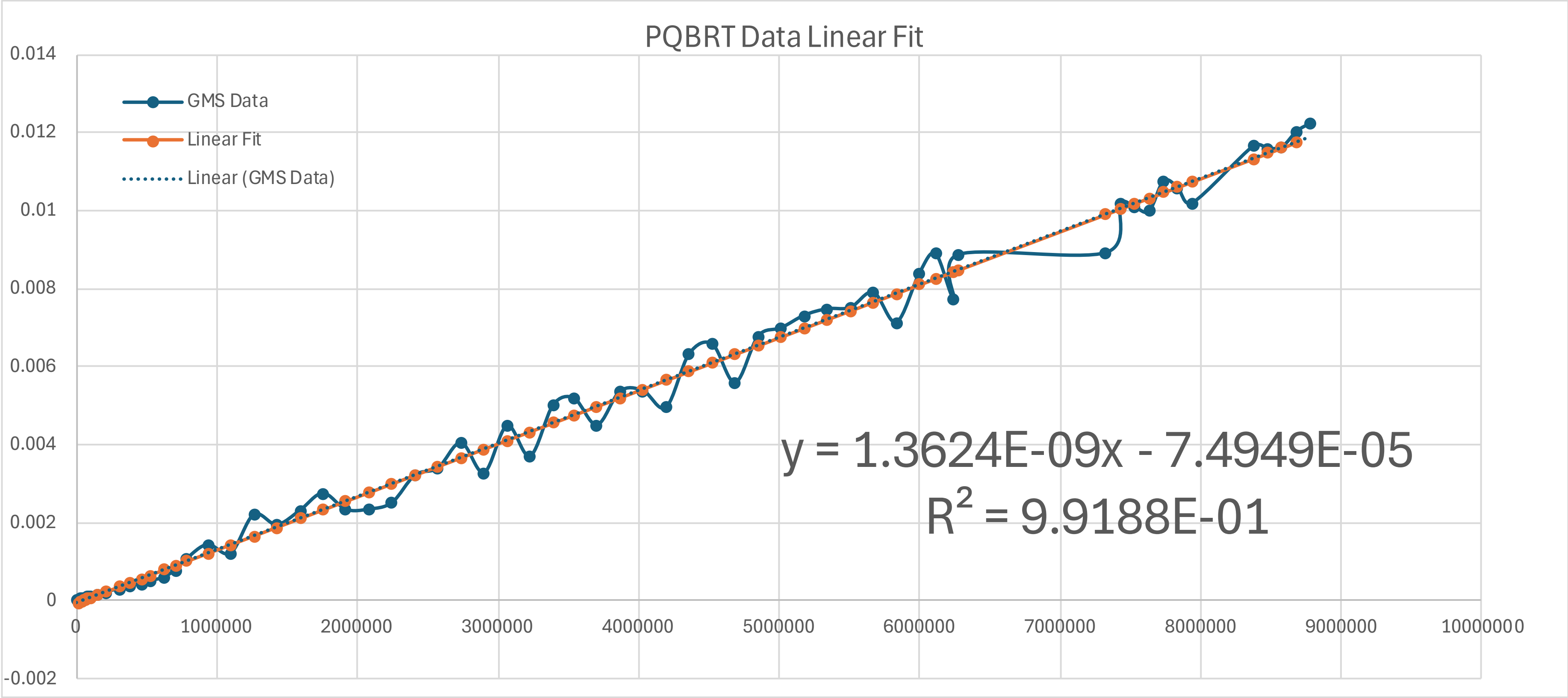
Graphics Pipeline Total Frame Time (spf)
Polynomial Fit

The residuals of the quadratic fit showing the mean in orange the data is fairly random around the mean while increasing with increasing time (increasing number of particles.



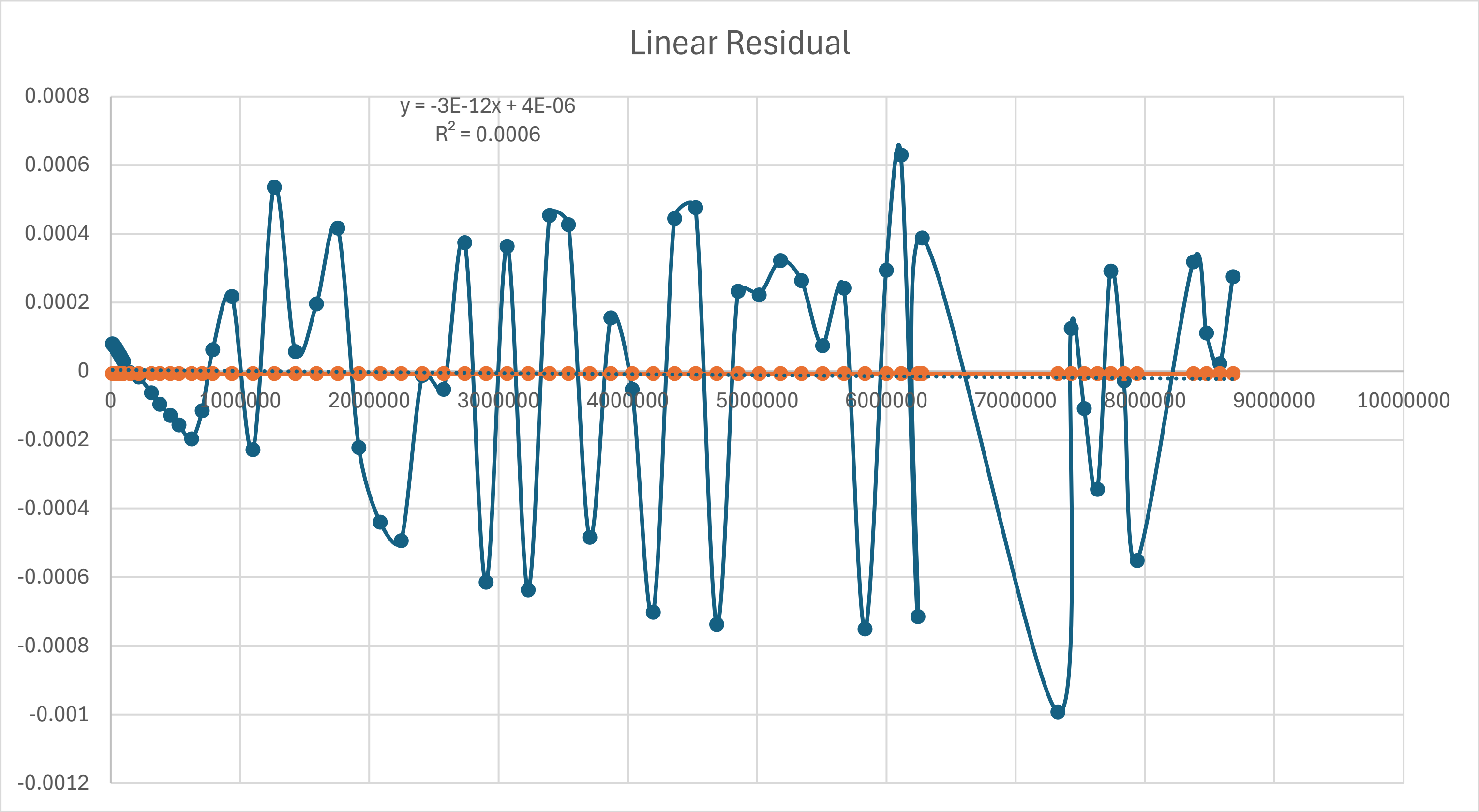
Graphics Pipeline Total Frame Time (spf)
Linear Fit

The quadratic trendline for the graphics pipeline. A completing algorithm has a base cost of 1.3624E-09 (s)



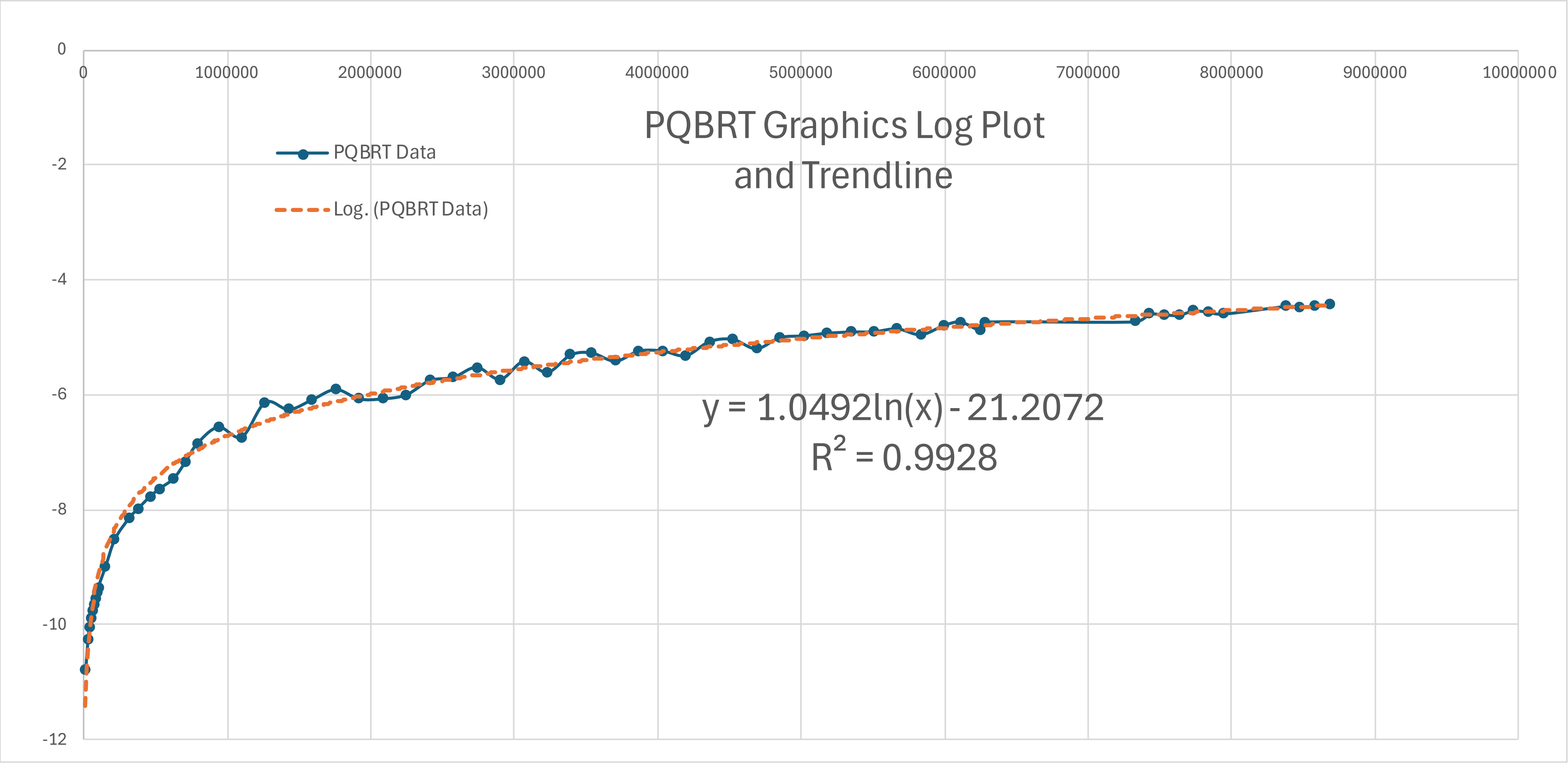
Graphics Pipeline Total Frame Time (spf)
Linear Fit

The residuals of the linear fit showing the mean in orange.



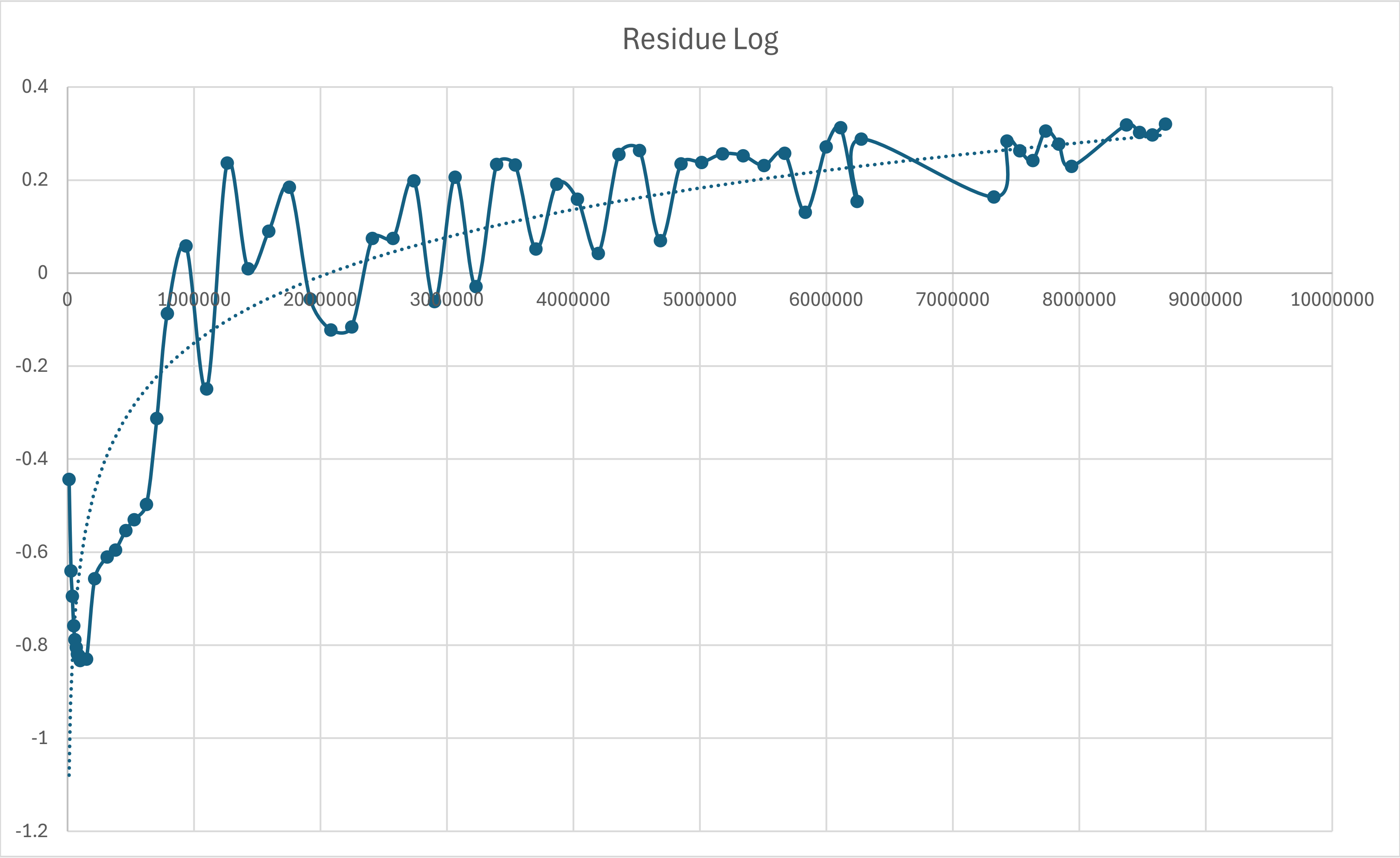
Graphics Pipeline Total Frame Time (spf)
Log plot

A log plot of graphics pipeline performance data. This is done to remove the **h**eteroscedasticity.



Graphics Pipeline Total Frame Time (spf)
Log plot

A plot of the log data residuals. This indicates that the variation in frame times increase while frame times increase. This accounts for the increase of the magnitude of the quadratic and linear plots variation.



The Compute Pipeline

Performance data interpretation