MGDO - Waveform Transforms

Michael Marino
MaGe Meeting, München
18 January 2009

Basics

- Transforms: anything that does a calculation using a waveform (may or may not modify wf)
- Design principal, same as OO: Do one thing, do it well. Transforms can be chained.

Basics

- MGVWaveformTransformer: base class for time-domain transforms
- MGVFreqDomainTransformer: base class for frequency-domain transforms (derives from above.) Automatically performs FFT on waveform.

Basics - For users

- Every transform has the following function:
- virtual void Transform(MGWaveform* input, MGWaveform* output);
- FFT: virtual void TransformFT(MGWaveformFT* input, MGWaveformFT* output=NULL); // This performs the transformation on FT-ed wfs. Useful if you don’t want to use the automatic transformation performed by the base class.
- NULL output means the function is performed “in-place” (input waveform is modified). This is ignored for pure calculation Transforms, i.e. those which

Basics - For coders

- Functions to overload (base class handles calling the correct function at run-time).
- virtual bool IsInPlace() = 0; // Overload this to tell base class if the transform does in-place transformations.
- virtual void TransformInPlace(MGWaveform& /*input*/) {} // Overload this if transformation is in-place.
- virtual void TransformOutOfPlace(const MGWaveform& /*intput*/, MGWaveform& /*output*/) {} // Overload this if transformation cannot be done in-place.
- Also, base class requires a name passed into the constructor, i.e.: MGVWaveformTransformer(const std::string& aTransformationName)

Basics - For coders

- FFT base class is similar:
- virtual bool IsFTInPlace() = 0; // Overload this to tell base class if the transform does in-place transformations.
- virtual void TransformFTInPlace(MGWaveformFT& /*input*/) {} // Overload this if transformation is in-place.
- virtual void TransformFTOutOfPlace(const MGWaveformFT& /*intput*/, MGWaveform& /*output*/) {} // Overload this if transformation cannot be done in-place.
- virtual void TransformFT(MGWaveformFT* input, MGWaveformFT* output=NULL); // This performs the transformation on FT-ed wfs. Useful if you don't want to use the automatic transformation.

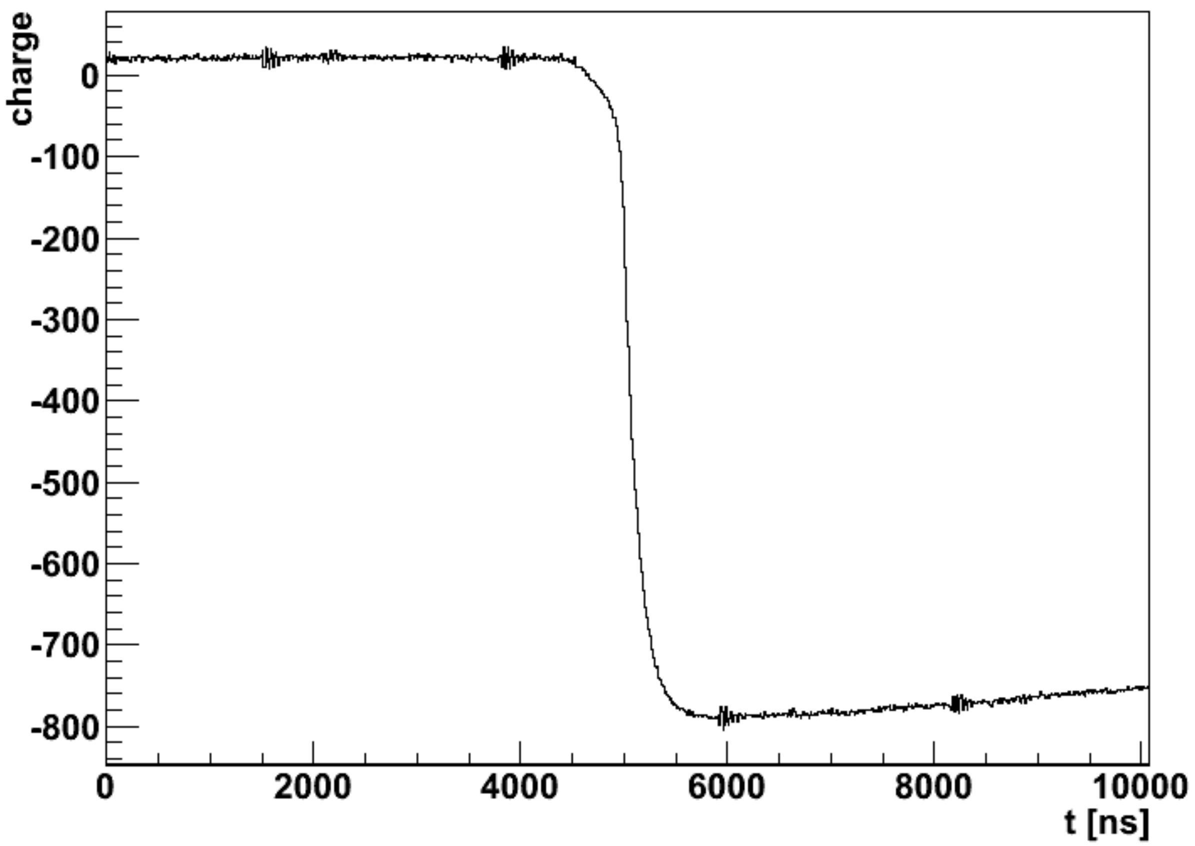
Utility Classes

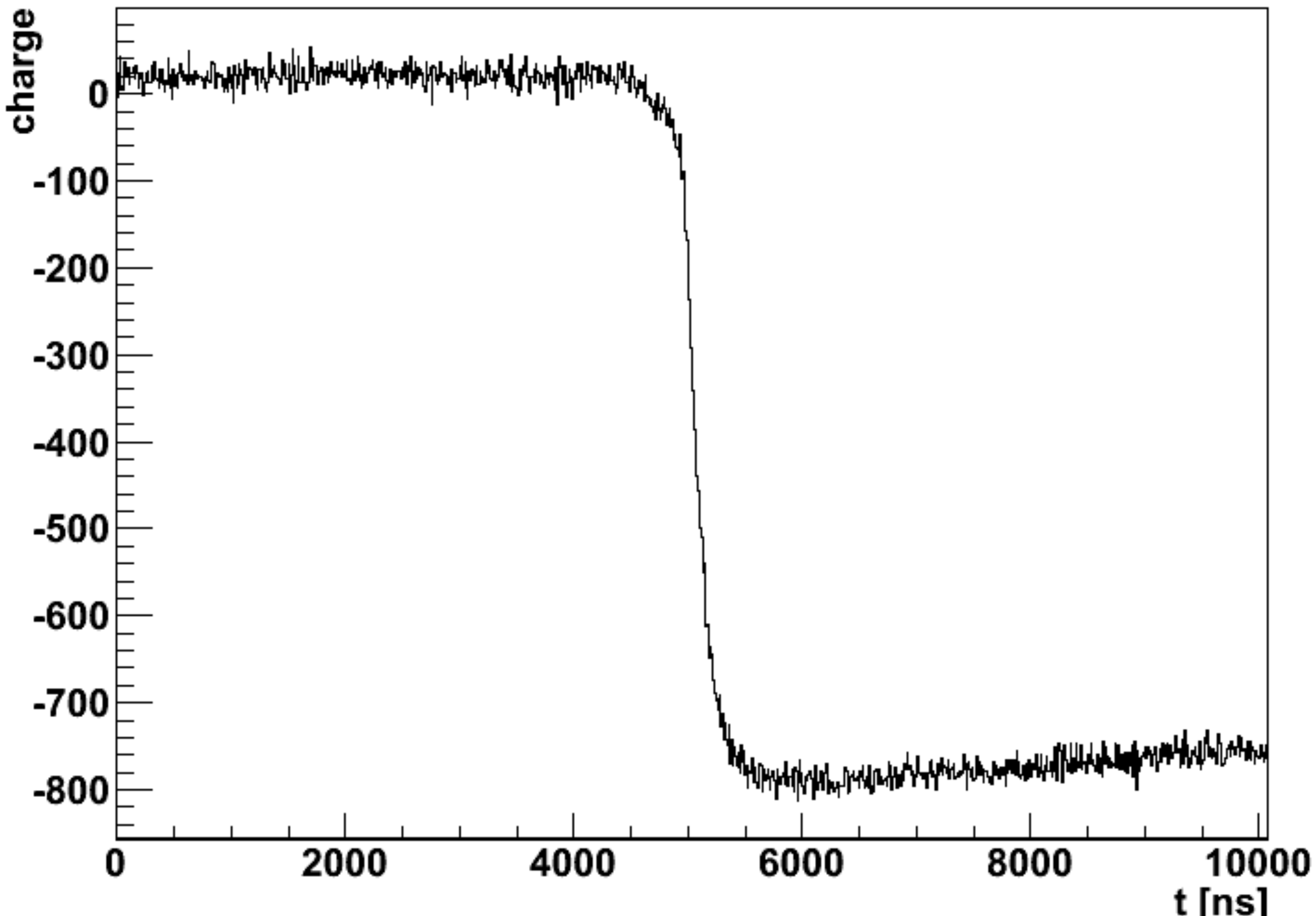
- MGWFMulti*Transformer
- Allows chaining together of transforms
- Call void AddWFTransformer(MGVWaveformTransformer*); then void Transform(MGWaveform* input, MGWaveform* output = NULL);
- Especially good for FT waveforms, performs FFT only once.

Available Transforms

MGWFAddNoise

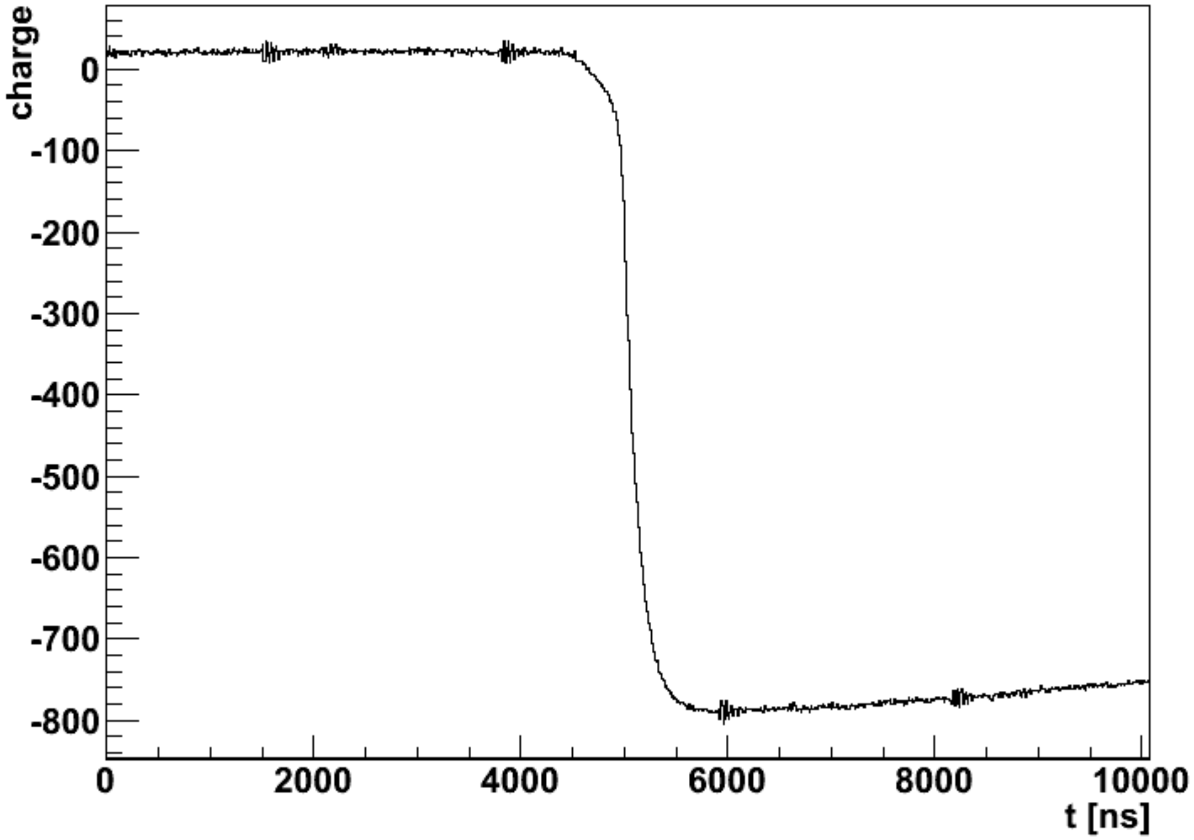
- Adds noise to a waveform (primitive, soon to be updated to use FT of noise).

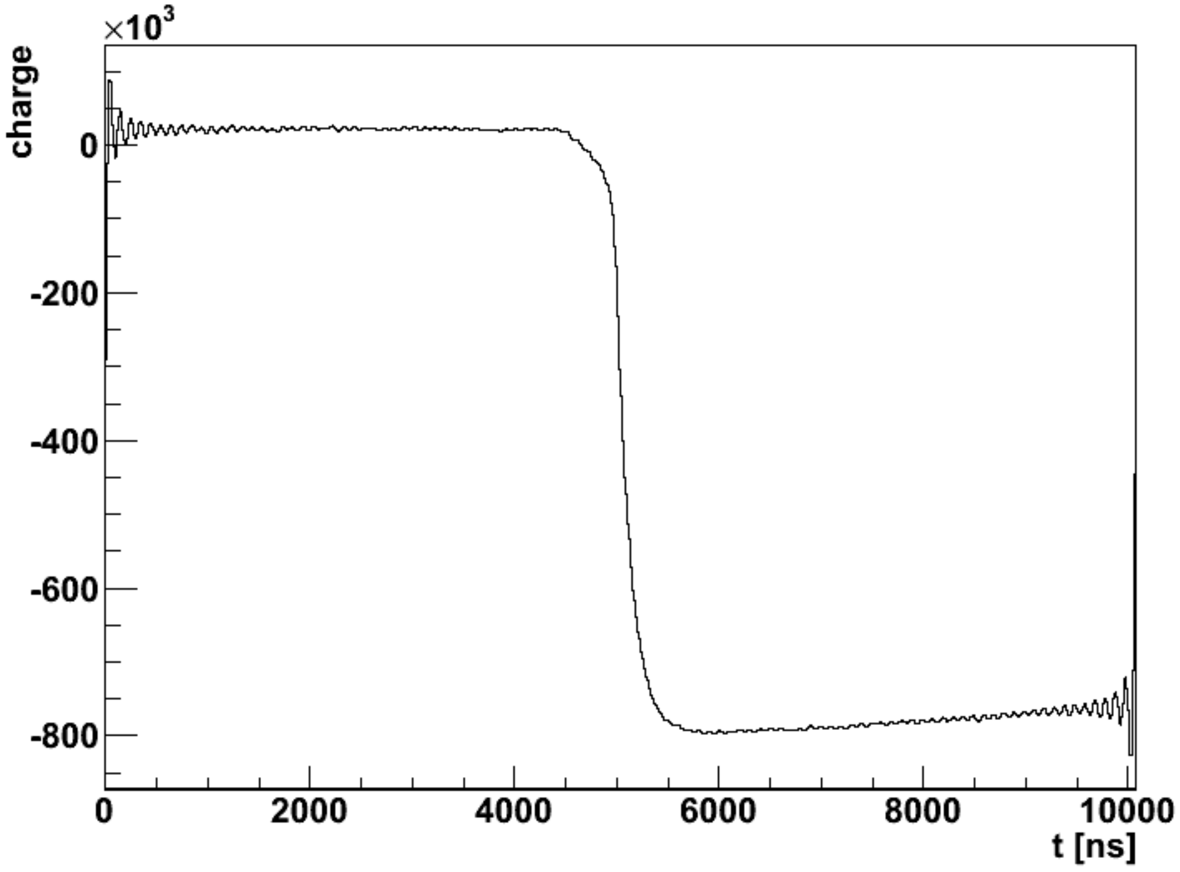




MGWFBandpassFilter

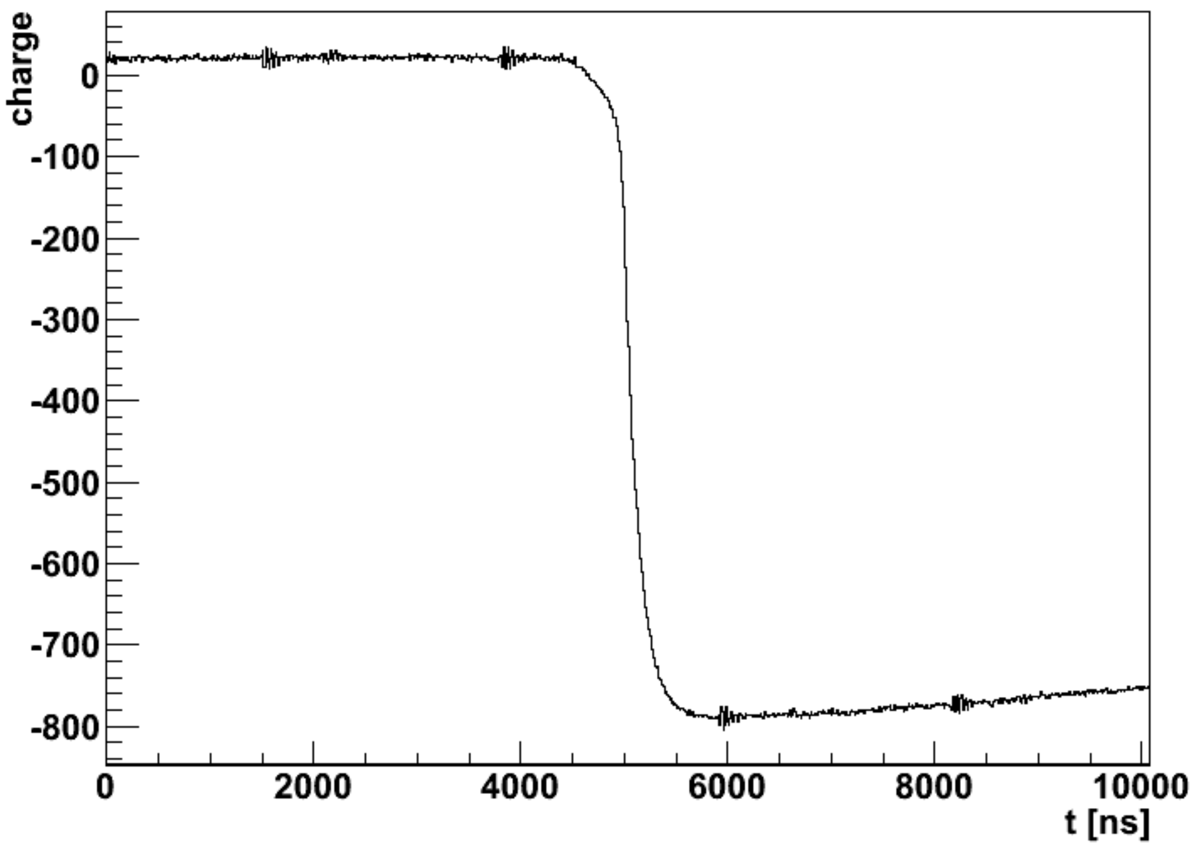
- Applies a bandpass to a waveform.

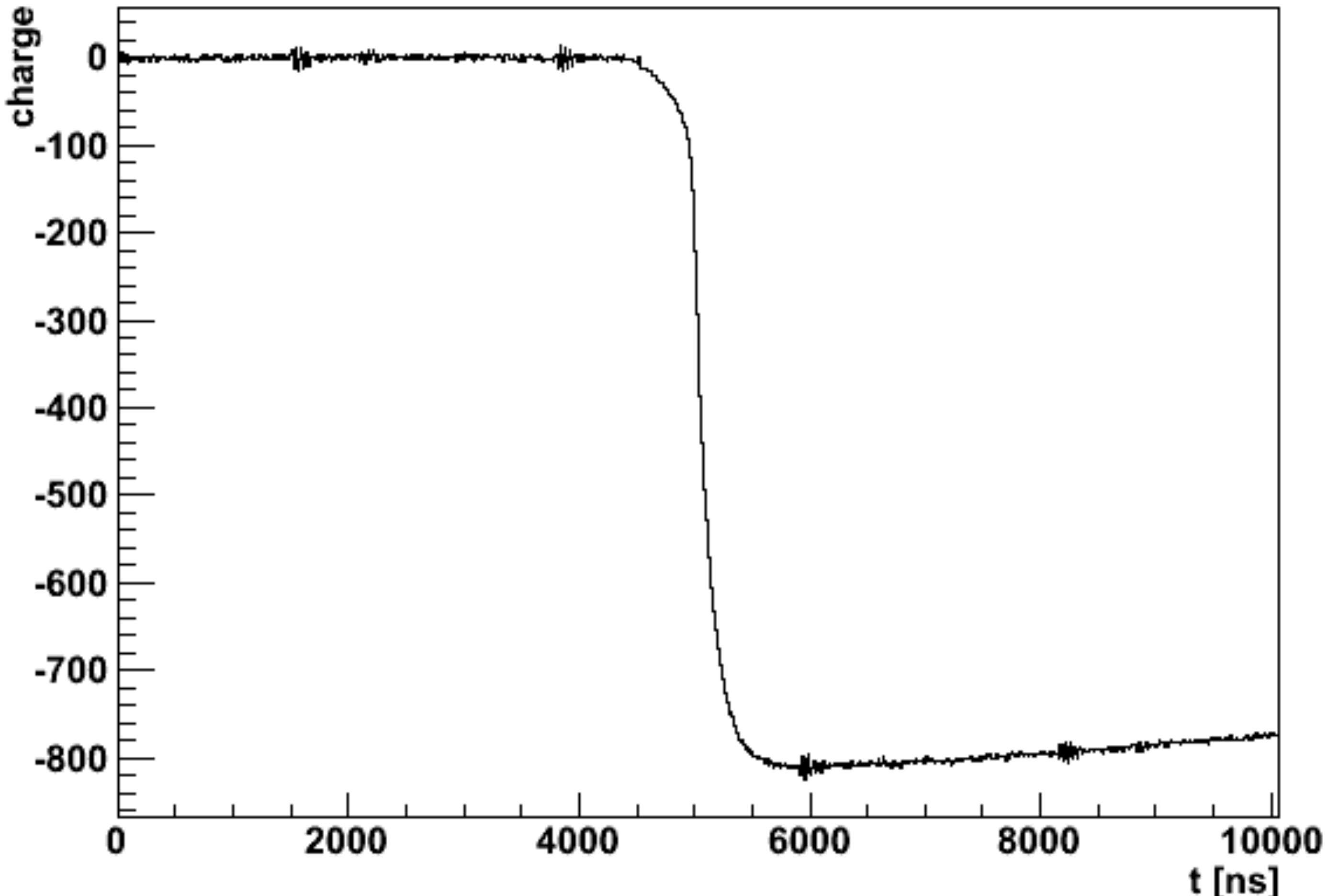




MGWFBaselineRemover

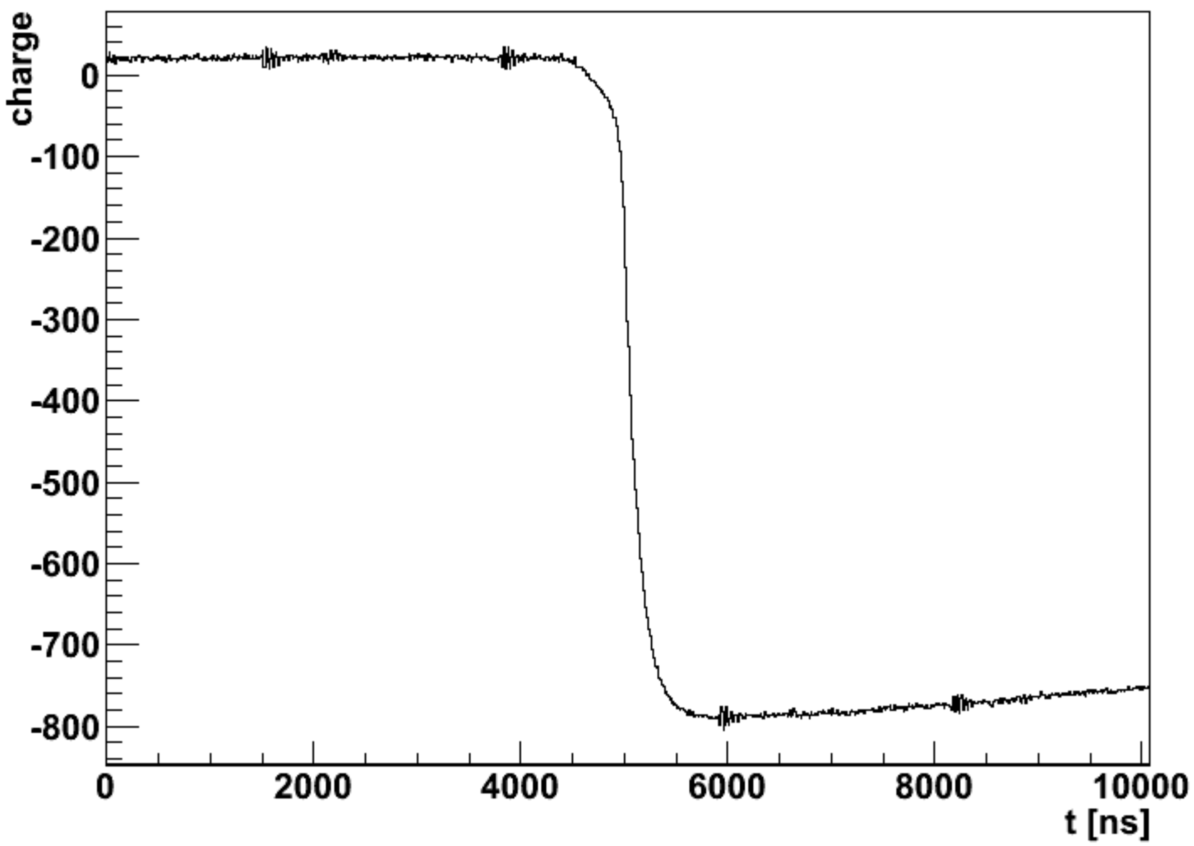
- Removes the baseline.

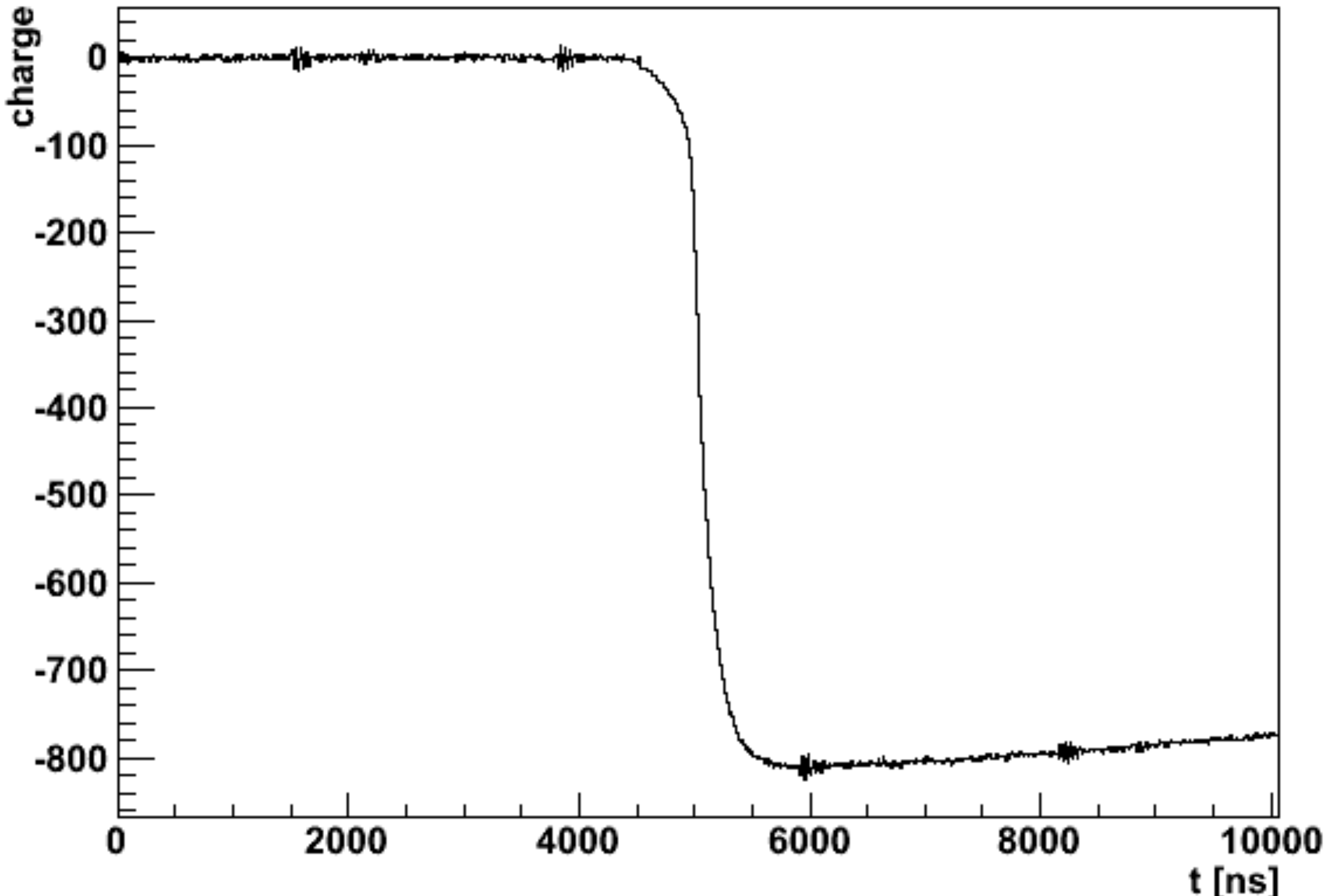




MGWFBaselineRemover

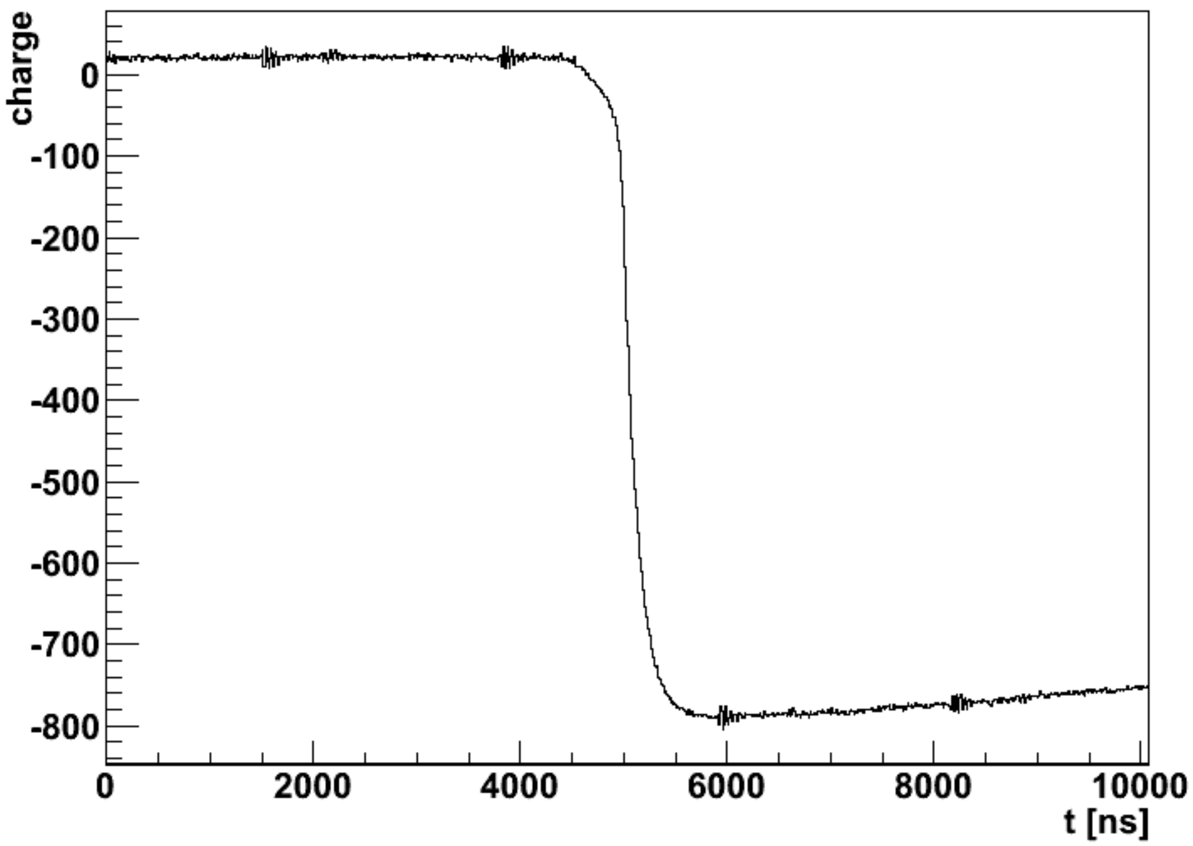
- Removes the baseline.

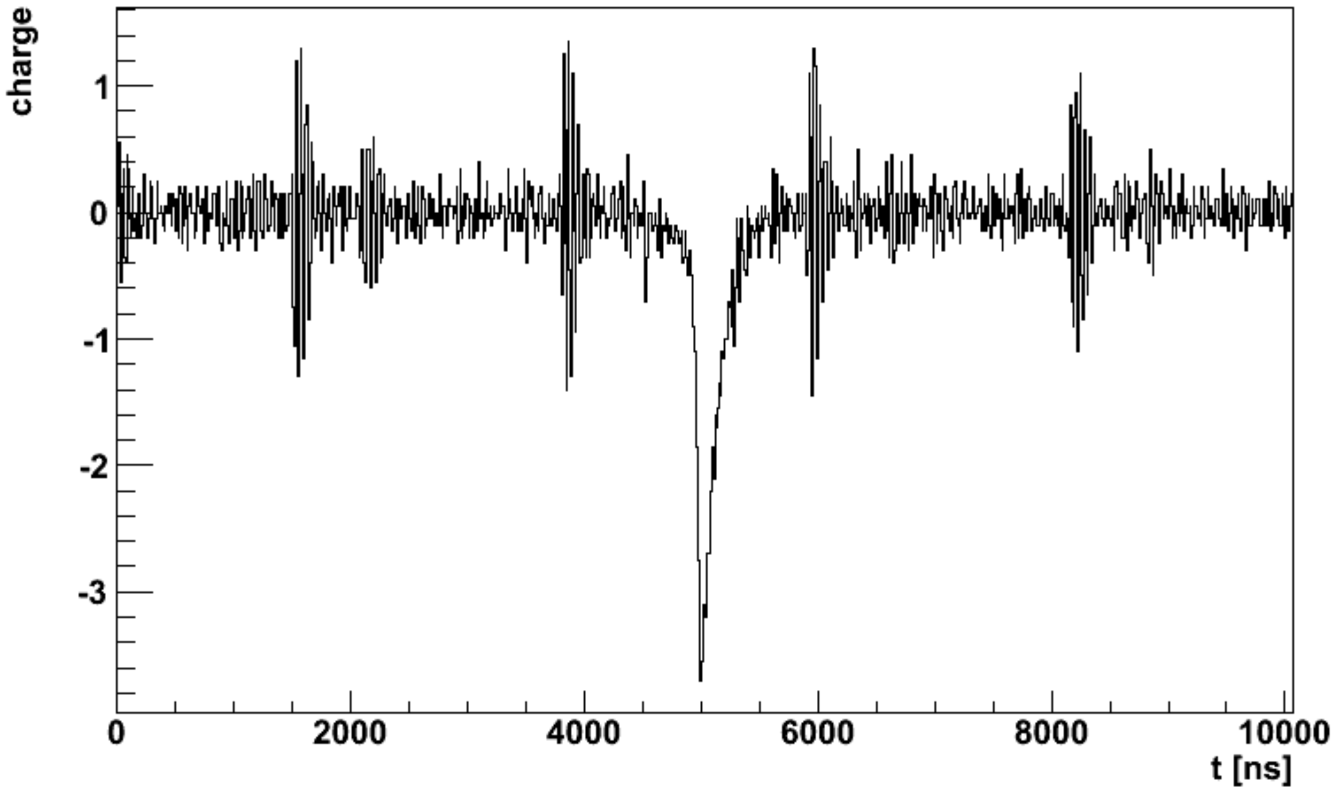




MGWFDerivative, MGWFDerivativeFourthOrder

- Takes the derivative





Live Demonstration

Available classes

- MGWFDigitizer
- Takes a "smooth" waveform with double precision and quantizes the amplitude. For PSA.
- MGWFExtremumFinder
- Find max or min of a waveform.
- MGWFIntegral
- Integrates class.
- MGWFSmoother, MGWFSavitzkyGolaySmoother
- Smoothing via S-G method.
- MGWFMovingAverage
- Performs moving average, smoothing

Available classes

- MGWFMovingWindow
- Performs moving average (e.g. used for trapezoidal filter)
- MGWFPoleZeroCorrection
- Corrects for pole-zero, assumes a baseline-removed waveform.
- MGWFPulseFinder
- Looks for a pulse that crosses a threshold (very simple threhold trigger).
- MGWFRCDifferentiation, MGWFRCIntegration
- Simple classes which mimic an RC preamp.
- MGWFResampler
- Transforms a waveform with one sampling frequency down to another with another frequency.
- Can choose interpolation methods.

Available classes

- MGWFRisetimeCalculation
- Calculates rise-time of pulses
- MGWFShiftSamples
- Shifts samples left or right.
- MGWFStaticWindow, MGWFStaticWindowCusp
- Static window functions basically perform moving window average at one point in the waveform.
- Used in energy calculation.