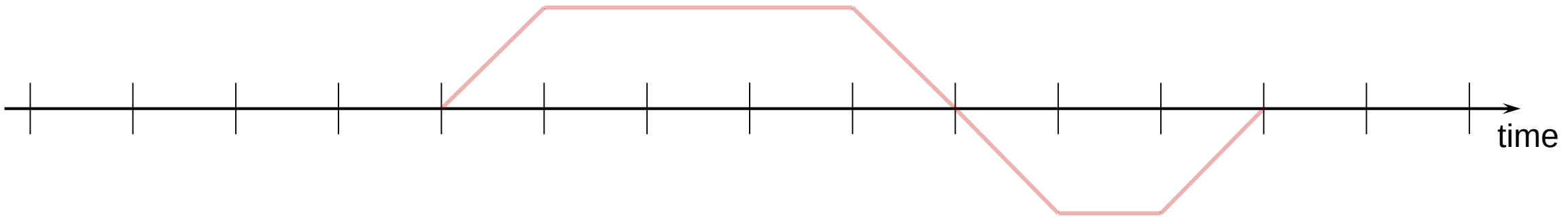
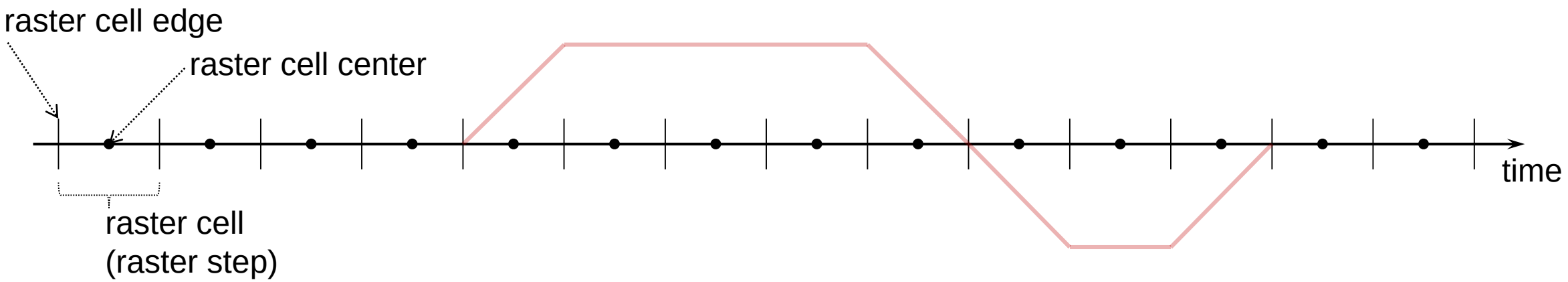
Pulseq
Time and Shape Specification

Shapes and Raster Times in *Pulseq*



- Specification needed for achieving precise control of gradient and RF waveforms (was missing prior to 1.4.0)
- Pulseq defines four types of raster times
  - adcRasterTime, rfRasterTime, gradRasterTime, blockDurationRaster
- Raster ‘thinking’ is probably one of the most demanding concepts in the practical pulse sequence programming
- Important concepts: **raster cells, edges and centers**

Definitions

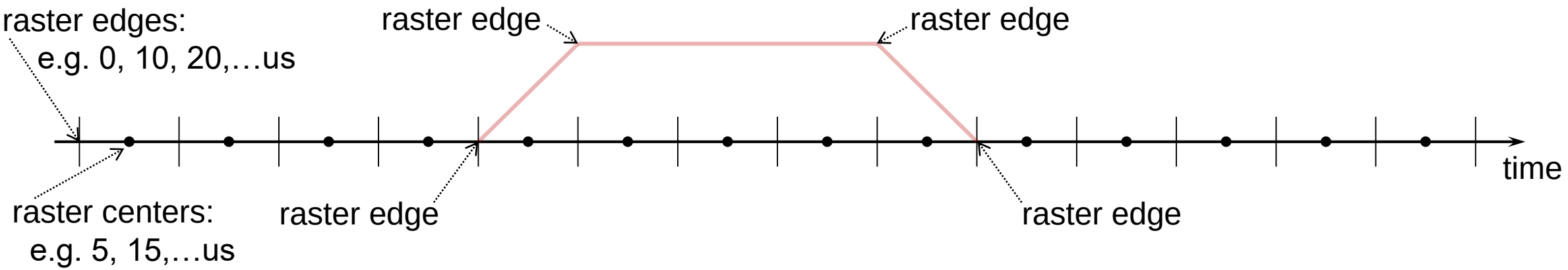


- The continuous and uninterrupted time axes is split in discrete time intervals, ***raster cells***, each of the equal duration of ***raster step***
- Each raster cell begins with the ***raster edge***
- The center of each raster cell is termed ***raster center***

Shapes in *Pulseq*

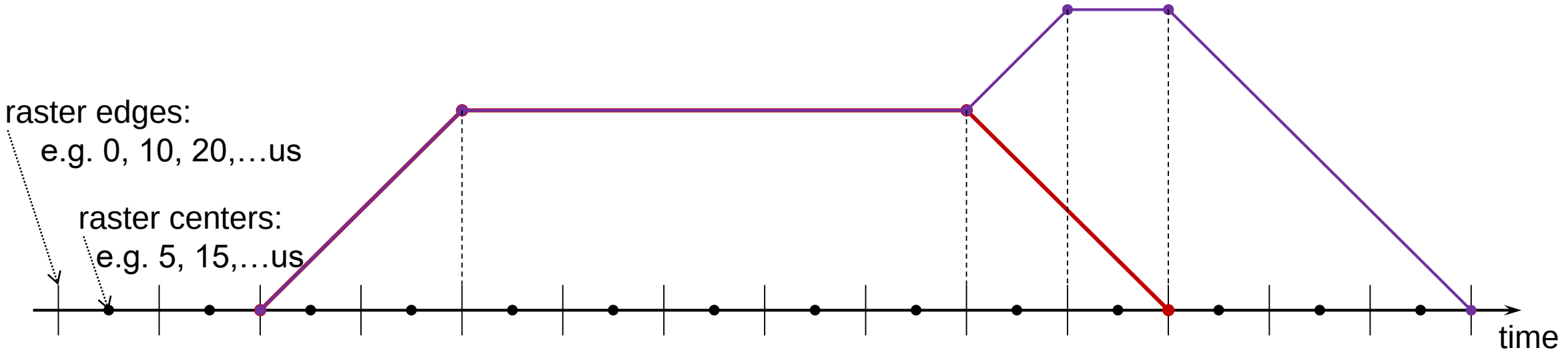
- Objects stored in [Shapes] are 1D vectors
  - Complex-valued shapes (e.g. for RF):two vectors (amplitude & phase)
- Two types of time representation:
  - Shapes with a regular sampling
    - No time_shape_id provided
    - Sampling time points are ALWAYS at raster cell centers
    - Time vector can be restored based on the corresponding raster step as ([1:N] - 0.5) * raster_step % Matlab notation
  - Shapes with explicit time vectors (time_shape_id provided)
    - Sampling points are (typically) on raster cell edges, but this is not a requirement

Conventional Trapezoid Gradient



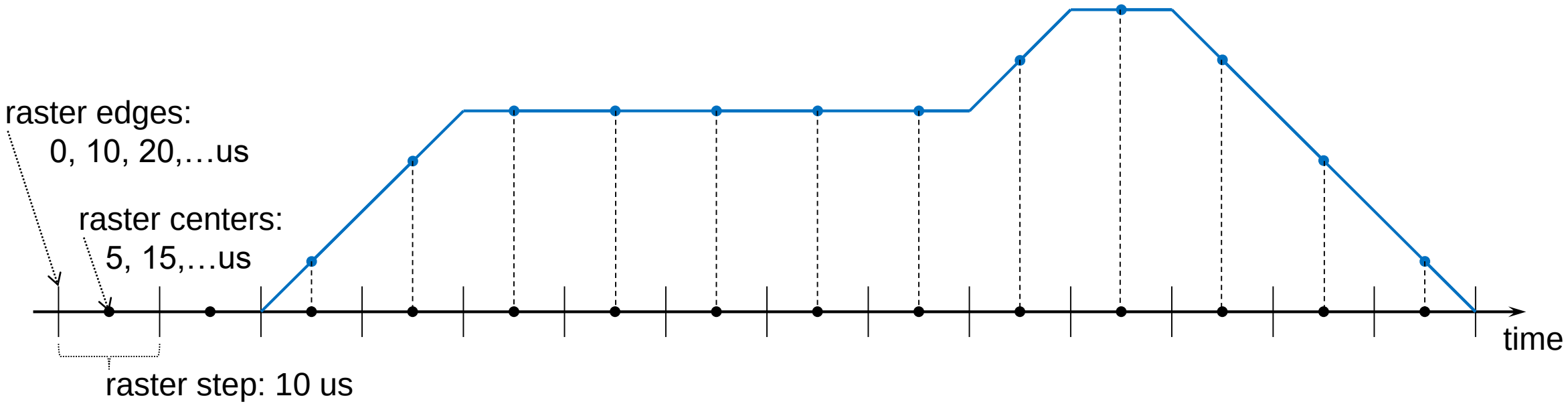
- Conventional gradients begin and end on gradient raster edges
- All timing elements of trapezoids need to be multiple of gradientRasterTime
- Beginning and end of the gradient flat top are both aligned to gradient raster edges
- The value of gradRasterTime (raster step) on Siemens is 10 us

Extended Trapezoid Gradient



- Conventional trapezoid has it vertices on raster edges
- Extended trapezoid is a generalization of a conventional trapezoid
- Arbitrary number of vertices, all aligned on raster edges
  - Implemented by providing time_shape_id with the “Grad” object
- Extended trapezoids may start/end at non-zero amplitude(s)
  - Vertices with non-zero amplitude must touch block boundary (to connect to neighbors)

Gradient with a Regularly-Sampled Shape



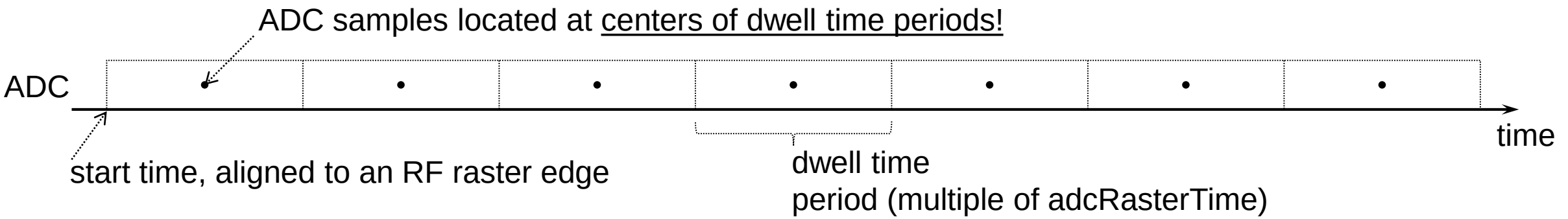
- Sampled (a.k.a. arbitrary) gradients: samples on raster centers
  - Gradient raster is 10 us on Siemens
  - Note raster differences to conventional and extended trapezoids!
- Implemented by providing time_shape_id with the “Grad” object
- As for extended trapezoids: non-zero start/end values are allowed

RF Raster Times

- RF objects can be either regularly sampled or defined by vertices
  - mr.makeSincPulse() defines a regularly sampled pulse
  - mr.makeBlockPulse() uses a shape with two points: (0,1) and (dur,1)
- The majority of shaped RF pulses are regularly sampled
  - Dwell time for regularly-sampled pulses: multiple of rfRasterTime
- Raster alignment rules:
  - The beginning of an RF object must be aligned to the RF raster edge
  - Sampling points of RF pulses are aligned to the centers of the dwell time periods
- rfRasterTime on Siemens: 1us

ADC Raster Times

- Special rules for the ADC (mixing RF and ADC raster times)
  - ADC start time must be aligned to rfRasterTime
  - ADC dwell time: multiple of adcRasterTime
- adcRaster time is 100ns on Siemens
- Sampling is assumed to happen instantly at the **centers** of the dwell time periods



Block Duration Raster

- blockDurationRaster is the atomic time step defining the duration of all blocks in a Pulseq sequence
  - All blocks have duration integer-multiple of blockDurationRaster

- All blocks begin and end at edges of the block raster
- Edges of all types of rasters coincide at the beginning of the block
  - rfRasterTime and gradRasterTime must be integer-multiples of blockDurationRaster or vise versa