

# Working with Pulseseq objects

Jon-Fredrik Nielsen

November 15, 2023

# Preliminaries

- Code for this PDF (Latex, MATLAB) available on course Github site
- Custom functions written for this talk end with '\_\_\_' suffix. Example:

```
>> sincrf___;           % creates example Pulseq events ('rf' and 'gz')
```

- Other functions belong to the Pulseq MATLAB toolbox, either as standalone functions or attached to a 'seq' object. Examples:

```
>> [rf, gz] = mr.makeSincPulse(pi/2, 'Duration', 2e-3, 'SliceThickness', 5e-3, ...  
    'apodization', 0.42, 'timeBwProduct', 4, 'system', sys);  
>> seq.plot('timeRange', [0 0.1]);
```

# Outline

- 1 Units
- 2 Gradient events
- 3 RF events
- 4 ADC events
- 5 Delay events
- 6 Creating blocks

# Units

- RF
  - ▶ Unit: Hz
  - ▶ divide by `sys.gamma` to convert to Tesla
- Gradients
  - ▶ Unit: Hz/m
  - ▶ divide by `sys.gamma` to convert to Tesla/m
- Time in seconds
- Flip angles in radians
- RF/ADC phase in radians

# Gradient events

3 types:

- Trapezoid
- Extended trapezoid
- Arbitrary gradient

Pulseq toolbox defaults

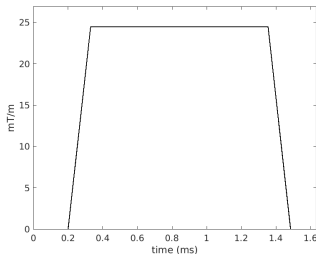
- `gradRasterTime` = 10e-6 s

# Trapezoid

Shape specified by rise/flat/fall times

```
Nx = 256;  
fov = 240e-3;      % m  
dwell = 4e-6;      % sec  
deltak = 1/fov;  
gx = mr.makeTrapezoid('x', 'FlatArea', Nx*deltak, 'FlatTime', Nx*dwell, ...  
                      'delay', 200e-6, 'system', sys);
```

```
gx =  
  type: 'trap'  
  channel: 'x'  
  amplitude: 1.0417e+06  
  riseTime: 1.3000e-04  
  flatTime: 0.0010  
  fallTime: 1.3000e-04  
  area: 1.2021e+03  
  flatArea: 1.0667e+03  
  delay: 2.0000e-04  
  first: 0  
  last: 0
```

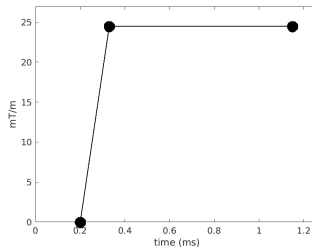


# Extended trapezoid

## Shape sampled on vertices

```
gxtmp = mr.makeTrapezoid('x', 'FlatArea', Nx*deltak, 'FlatTime', Nx*dwell, ...  
                          'delay', 200e-6, 'system', sys);  
[gx,~] = mr.splitGradientAt(gxtmp, gxtmp.riseTime + gxtmp.flatTime);  
  
% Or:  
gx = mr.makeExtendedTrapezoid('x', 'times', gx.tt, 'amplitudes', gx.waveform);  
gx.delay = 200e-6;
```

```
gx =  
  type: 'grad'  
  channel: 'x'  
  waveform: [0 1.0417e+06 1.0417e+06]  
  delay: 2.0000e-04  
  tt: [0 1.3000e-04 9.5000e-04]  
  shape_dur: 9.5000e-04  
  area: 921.8750  
  first: 0  
  last: 1.0417e+06
```



# Arbitrary gradient

Shape sampled on regular raster, on **center** of raster periods

```
spiral; % creates 'wav' vector, mT/m
gx = mr.makeArbitraryGrad('x', wav*sys.gamma/1e3, 'delay', 200e-6);
```

```
gx =
  type: 'grad'
  channel: 'x'
  waveform: [148.3082 1.3283e+03 3.6538e+03 7.0567e+03 1.1437e+04 1.6663e+04 2.2579e+04 2.
  delay: 2.0000e-04
  area: -42.3091
  tt: [5.0000e-06 1.5000e-05 2.5000e-05 3.5000e-05 4.5000e-05 5.5000e-05 6.5000e-05
  shape_dur: 0.0010
  first: -441.6805
  last: -666.5979
```

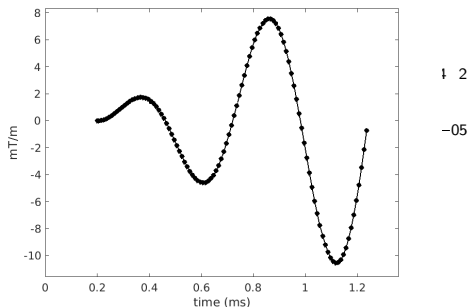


# Arbitrary gradient

Shape sampled on regular raster, on **center** of raster periods

```
spiral; % creates 'wav' vector, mT/m  
gx = mr.makeArbitraryGrad('x', wav*sys.gamma/1e3, 'delay', 200e-6);
```

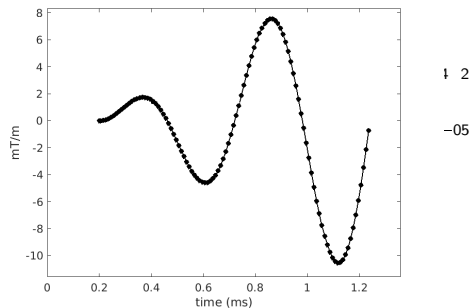
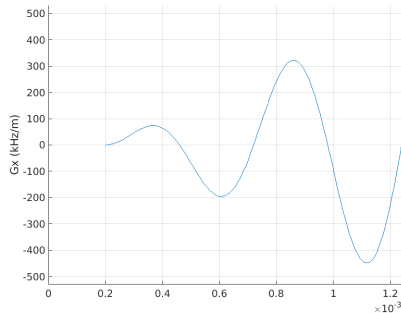
```
gx =  
  type: 'grad'  
  channel: 'x'  
  waveform: [148.3082 1.3283e+03 3.6538e+03]  
  delay: 2.0000e-04  
  area: -42.3091  
  tt: [5.0000e-06 1.5000e-05 2.5000e-05]  
  shape_dur: 0.0010  
  first: -441.6805  
  last: -666.5979
```



# Arbitrary gradient

Shape sampled on regular raster, on **center** of raster periods

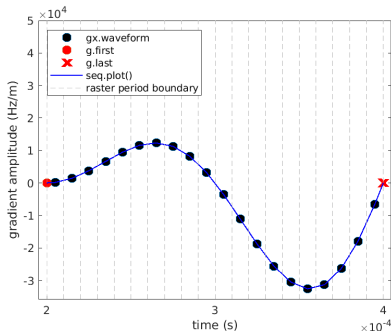
```
spiral; % creates 'wav' vector, mT/m  
gx = mr.makeArbitraryGrad('x', wav*sys.gamma/1e3, 'delay', 200e-6);
```



## 'first' and 'last'

- Waveform values at beginning of first raster period, and end of last raster period
- Calculated by `makeArbitraryGradient()`, or set manually
- Uses: `seq.plot()`; k-space calculation; check for waveform continuity across blocks
- not yet saved in `.seq` file (likely to change in the future)

```
gx =  
    type: 'grad'  
    channel: 'x'  
    waveform: [166.5088 1.4497e+03 3.7632e+03 6.  
    delay: 2.0000e-04  
    area: -1.3608  
    tt: [5.0000e-06 1.5000e-05 2.5000e-05  
    shape_dur: 2.0000e-04  
    first: 0  
    last: 0
```



# Modify gradients with scaleGrad()

- Function returns a scaled copy of the input event
- Typically used inside scan loop, e.g., to scale the y phase encode gradient:

```
for j = -Ny/2:Ny/2-1
    seq.addBlock(mr.scaleGrad(gy, j/(Ny/2)));
...
end
```

- Opinion: it is good programming practice use scaleGradient() instead of creating a new event from scratch
  - ▶ Makes the purpose of the gradient event, and relationship to the input gradient, clear
  - ▶ Scaled events share the same shape ID → simplifies interpreter code

# Modify gradients with rotate()

- Function returns single-axis gradient events resulting from rotation

```
gx =
  type: 'grad'
  channel: 'x'
  waveform: [166.5088 1.4497e+03 3.7632e+03 6.6387e+03 9.4617e+03 1.1564e+04 1.2329e+04]
  delay: 2.0000e-04
  area: -1.3608
  tt: [5.0000e-06 1.5000e-05 2.5000e-05 3.5000e-05 4.5000e-05 5.5000e-05 6.5000e-05]
  shape_dur: 2.0000e-04

>> [gxr gyr] = mr.rotate('z', pi/4, gx)
gxr =
  type: 'grad'
  channel: 'x'
  waveform: [117.7395 1.0251e+03 2661 4.6943e+03 6.6904e+03 8.1772e+03 8.7179e+03 7.1779e+03]
  delay: 2.0000e-04
  area: -1.3608
  tt: [5.0000e-06 1.5000e-05 2.5000e-05 3.5000e-05 4.5000e-05 5.5000e-05 6.5000e-05]
  shape_dur: 2.0000e-04

gyr =
  type: 'grad'
  channel: 'y'
  waveform: [117.7395 1.0251e+03 2.6610e+03 4.6943e+03 6.6904e+03 8.1772e+03 8.7179e+03 7.1779e+03]
  delay: 2.0000e-04
  area: -1.3608
  tt: [5.0000e-06 1.5000e-05 2.5000e-05 3.5000e-05 4.5000e-05 5.5000e-05 6.5000e-05]
  shape_dur: 2.0000e-04
```

# Gradient 'surgery' with splitGradientAt()

```
gyBlip =  
  type: 'trap'  
  channel: 'y'  
  amplitude: 5.5556e+05  
  riseTime: 9.0000e-05  
  flatTime: 0  
  fallTime: 9.0000e-05  
  area: 50  
  
>> gy-parts = mr.splitGradientAt(gyBlip, gyBlip.riseTime);  
gy-parts(1) =  
  type: 'grad'  
  channel: 'y'  
  waveform: [0 5.5556e+05]  
  delay: 0  
  tt: [0 9.0000e-05]  
  shape_dur: 9.0000e-05  
  area: 25  
  
gy-parts(2) =  
  type: 'grad'  
  channel: 'y'  
  waveform: [5.5556e+05 0]  
  delay: 9.0000e-05  
  tt: [0 9.0000e-05]  
  shape_dur: 9.0000e-05  
  area: 25
```

# addGradients()

Returns sum of gradients

```
gy =  
    type: 'trap'  
    channel: 'y'  
    amplitude: 8.3333e+05  
    riseTime: 1.2000e-04  
    flatTime: 0  
    fallTime: 1.2000e-04  
    area: 100  
    flatArea: 0  
    delay: 0  
  
>> gy_sum = mr.addGradients({gy, gy}, 'system', sys)  
  
gy_sum =  
    type: 'trap'  
    channel: 'y'  
    amplitude: 1.6667e+06  
    riseTime: 1.2000e-04  
    flatTime: 0  
    fallTime: 1.2000e-04  
    area: 200  
    flatArea: 0  
    delay: 0
```

# RF events

- Shape sampled on regular raster on center of raster periods, or on vertices
- Pulseseq toolbox defaults
  - ▶ `rfRasterTime = 1e-6 s`



# RF event: Example 1

## sinc pulse

```
[rf, gz] = mr.makeSincPulse(20/180*pi, 'Duration', 2e-3, 'SliceThickness', 5e-3, ...  
    'apodization', 0.42, 'timeBwProduct', 4, 'system', sys);
```

```
rf =  
    type: 'rf'  
    signal: [-0.0089 -0.0268 -0.0448 -0.0628 -0.0808 -0.0988 -0.1169 -0.1350 -0.1532 ...  
    t: [5.0000e-07 1.5000e-06 2.5000e-06 3.5000e-06 4.5000e-06 5.5000e-06 6.5000e-06 ...  
    shape_dur: 0.0020  
    freqOffset: 0  
    phaseOffset: 0  
    deadTime: 1.0000e-04  
    ringdownTime: 6.0000e-05  
    delay: 1.0000e-04  
  
gx =  
    type: 'grad'  
    channel: 'x'  
    waveform: [166.5088 1.4497e+03 3.7632e+03 6.6387e+03 9.4617e+03 1.1564e+04 1.2329e+04 1 ...  
    delay: 2.0000e-04  
    area: -1.3608  
    tt: [5.0000e-06 1.5000e-05 2.5000e-05 3.5000e-05 4.5000e-05 5.5000e-05 6.5000e-05 ...  
    shape_dur: 2.0000e-04  
    first: 0  
    last: 0
```

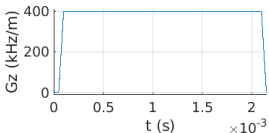
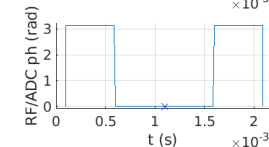
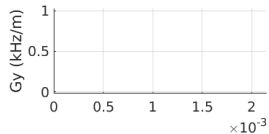
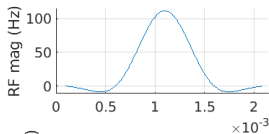
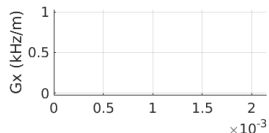
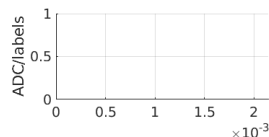
# RF event: Example 1

## sinc pulse

```
[rf, gz] = mr.makeSincPulse(20/180*pi, 'Duration', 2e-3, 'SliceThickness', 5e-3, ...  
    'apodization', 0.42, 'timeBwProduct', 4, 'system', sys);
```

```
rf =  
    type: 'rf'  
    signal: [-0.0089 -0.02  
            t: [5.0000e-07 1.  
    shape_dur: 0.0020  
    freqOffset: 0  
    phaseOffset: 0  
    deadTime: 1.0000e-04  
    ringdownTime: 6.0000e-05  
    delay: 1.0000e-04
```

```
gx =  
    type: 'grad'  
    channel: 'x'  
    waveform: [166.5088 1.4497e  
    delay: 2.0000e-04  
    area: -1.3608  
    tt: [5.0000e-06 1.500  
    shape_dur: 2.0000e-04  
    first: 0  
    last: 0
```



## RF dead time, ringdown time

- Some time is required to turn on RF amplifier
  - ▶ Varies by vendor (of order  $\sim 100 \mu s$ )
- Some time is also required to turn off the RF amplifier
  - ▶ Varies by vendor (of order tens of  $\mu s$ )
- Pulseq toolbox:
  - ▶ Requires that `rf.delay`  $\geq$  `sys.rfDeadTime`. `makeSincPulse()` will insert this delay if needed.
  - ▶ Requires a gap  $\geq$  `sys.rfRingdownTime` between end of RF waveform and end of block. `makeSincPulse()` will extend the block duration if needed.
- Depending on the vendor, it may not always be necessary to specify non-zero dead/ringdown times (more on that later)

## RF event: Example 2

### SMS pulse (6 simultaneous slices)

```
load arbrf.mat;    % rfwav (T) and gzwav (mT/m)

flip = pi;    % dummy value, will rescale anyhow
rf = mr.makeArbitraryRf(rfwav, flip, 'delay', sys.rfDeadTime, 'system', sys);
rf.signal = rf.signal/max(abs(rf.signal))*max(abs(rfwav)*sys.gamma); % Hz

gzwav_Hzm = gzwav*sys.gamma/1e3;
gz = mr.makeArbitraryGrad('z', gzwav_Hzm, sys, 'delay', 10e-6);
```

```
rf =
    type: 'rf'
    signal: [0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i]
    t: [5.0000e-07 1.5000e-06 2.5000e-06 3.5000e-06 4.5000e-06 5.5000e-06 6.5000e-06]
    shape_dur: 0.0080
    freqOffset: 0
    phaseOffset: 0
    deadTime: 1.0000e-04
    ringdownTime: 6.0000e-05
    delay: 1.0000e-04

gx =
    type: 'grad'
    channel: 'x'
    waveform: [166.5088 1.4497e+03 3.7632e+03 6.6387e+03 9.4617e+03 1.1564e+04 1.2329e+04 1.2329e+04]
    delay: 2.0000e-04
    area: -1.3608
    tt: [5.0000e-06 1.5000e-05 2.5000e-05 3.5000e-05 4.5000e-05 5.5000e-05 6.5000e-05]
    shape_dur: 2.0000e-04
    first: 0
    last: 0
```

# RF event: Example 2

## SMS pulse (6 simultaneous slices)

```
load arbrf.mat;      % rfwav (T) and gzwav (mT/m)
```

```
flip = pi; % dummy value, will rescale anyhow
```

```
rf = mr.makeArbitraryRf(rfwav, flip, 'delay', sys.rfDeadTime, 'system', sys);
```

```
rf.signal = rf.signal/max(abs(rf
```

```
gzwav.Hzm = gzwav*sys.gamma/1e3;
```

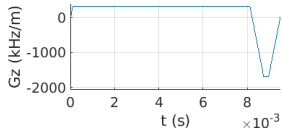
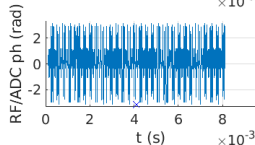
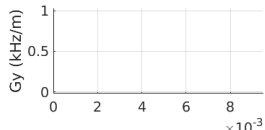
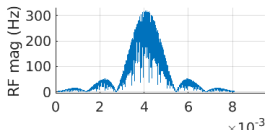
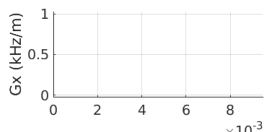
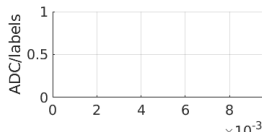
```
gz = mr.makeArbitraryGrad('z', g
```

```
rf =
```

```
    type: 'rf'  
  signal: [0.0000 + 0.00  
           t: [5.0000e-07 1..  
  shape_dur: 0.0080  
  freqOffset: 0  
  phaseOffset: 0  
    deadTime: 1.0000e-04  
  ringdownTime: 6.0000e-05  
    delay: 1.0000e-04
```

```
gx =
```

```
    type: 'grad'  
  channel: 'x'  
  waveform: [166.5088 1.4497e  
            delay: 2.0000e-04  
            area: -1.3608  
            tt: [5.0000e-06 1.500  
  shape_dur: 2.0000e-04  
    first: 0  
    last: 0
```



# ADC events

- Samples assumed to occur at centers of dwell time periods
- The delay defines the timing of the starting edge of the first period

```
Nx = 256;  
dwell = 4e-6; % sec  
adc = mr.makeAdc(Nx, 'Duration', Nx*dwell, 'Delay', 0.2e-3, 'system', sys);
```

```
adc =  
    type: 'adc'  
  numSamples: 256  
    dwell: 4.0000e-06  
    delay: 2.0000e-04  
  freqOffset: 0  
  phaseOffset: 0  
    deadTime: 4.0000e-05
```

# ADC dead time

- Some time is required to turn on data acquisition board
  - ▶ Varies by vendor (typically a few tens of  $\mu s$ )
- Pulseseq toolbox:
  - ▶ Requires that `adc.delay`  $\geq$  `sys.adcDeadTime`. `makeAdc()` will insert this delay if needed.
  - ▶ Requires a gap  $\geq$  `sys.adcDeadTime` between end of ADC window and end of block. `makeAdc()` will extend the block duration if needed.
- Depending on the vendor, it may not always be necessary to specify a non-zero dead time before/after the ADC window

# Delay events

- Not real objects: their only function is to (possibly) extend the block duration
- Not stored in the Pulseq file

```
del = mr.makeDelay(2e-3);
```

```
delay =  
  type: 'delay '  
  delay: 0.0020
```



## Get event duration: `mr.calcDuration()`

- Example:

```
rf =  
    type: 'rf'  
    signal: [-0.0089 -0.0268 -0.0448 -0.0628 -0.0808 -0.0988 -0.1169 -0.1350 -0.1531 -0.1712 -0.1893 -0.2074 -0.2255 -0.2436 -0.2617 -0.2798 -0.2979 -0.3160 -0.3341 -0.3522 -0.3703 -0.3884 -0.4065 -0.4246 -0.4427 -0.4608 -0.4789 -0.4970 -0.5151 -0.5332 -0.5513 -0.5694 -0.5875 -0.6056 -0.6237 -0.6418 -0.6599 -0.6780 -0.6961 -0.7142 -0.7323 -0.7504 -0.7685 -0.7866 -0.8047 -0.8228 -0.8409 -0.8590 -0.8771 -0.8952 -0.9133 -0.9314 -0.9495 -0.9676 -0.9857 -1.0038 -1.0219 -1.0400 -1.0581 -1.0762 -1.0943 -1.1124 -1.1305 -1.1486 -1.1667 -1.1848 -1.2029 -1.2210 -1.2391 -1.2572 -1.2753 -1.2934 -1.3115 -1.3296 -1.3477 -1.3658 -1.3839 -1.4020 -1.4201 -1.4382 -1.4563 -1.4744 -1.4925 -1.5106 -1.5287 -1.5468 -1.5649 -1.5830 -1.6011 -1.6192 -1.6373 -1.6554 -1.6735 -1.6916 -1.7097 -1.7278 -1.7459 -1.7640 -1.7821 -1.8002 -1.8183 -1.8364 -1.8545 -1.8726 -1.8907 -1.9088 -1.9269 -1.9450 -1.9631 -1.9812 -2.0000 -2.0181 -2.0362 -2.0543 -2.0724 -2.0905 -2.1086 -2.1267 -2.1448 -2.1629 -2.1810 -2.1991 -2.2172 -2.2353 -2.2534 -2.2715 -2.2896 -2.3077 -2.3258 -2.3439 -2.3620 -2.3801 -2.3982 -2.4163 -2.4344 -2.4525 -2.4706 -2.4887 -2.5068 -2.5249 -2.5430 -2.5611 -2.5792 -2.5973 -2.6154 -2.6335 -2.6516 -2.6697 -2.6878 -2.7059 -2.7240 -2.7421 -2.7602 -2.7783 -2.7964 -2.8145 -2.8326 -2.8507 -2.8688 -2.8869 -2.9050 -2.9231 -2.9412 -2.9593 -2.9774 -2.9955 -3.0136 -3.0317 -3.0498 -3.0679 -3.0860 -3.1041 -3.1222 -3.1403 -3.1584 -3.1765 -3.1946 -3.2127 -3.2308 -3.2489 -3.2670 -3.2851 -3.3032 -3.3213 -3.3394 -3.3575 -3.3756 -3.3937 -3.4118 -3.4299 -3.4480 -3.4661 -3.4842 -3.5023 -3.5204 -3.5385 -3.5566 -3.5747 -3.5928 -3.6109 -3.6290 -3.6471 -3.6652 -3.6833 -3.7014 -3.7195 -3.7376 -3.7557 -3.7738 -3.7919 -3.8100 -3.8281 -3.8462 -3.8643 -3.8824 -3.9005 -3.9186 -3.9367 -3.9548 -3.9729 -3.9910 -4.0091 -4.0272 -4.0453 -4.0634 -4.0815 -4.0996 -4.1177 -4.1358 -4.1539 -4.1720 -4.1901 -4.2082 -4.2263 -4.2444 -4.2625 -4.2806 -4.2987 -4.3168 -4.3349 -4.3530 -4.3711 -4.3892 -4.4073 -4.4254 -4.4435 -4.4616 -4.4797 -4.4978 -4.5159 -4.5340 -4.5521 -4.5702 -4.5883 -4.6064 -4.6245 -4.6426 -4.6607 -4.6788 -4.6969 -4.7150 -4.7331 -4.7512 -4.7693 -4.7874 -4.8055 -4.8236 -4.8417 -4.8598 -4.8779 -4.8960 -4.9141 -4.9322 -4.9503 -4.9684 -4.9865 -5.0046 -5.0227 -5.0408 -5.0589 -5.0770 -5.0951 -5.1132 -5.1313 -5.1494 -5.1675 -5.1856 -5.2037 -5.2218 -5.2399 -5.2580 -5.2761 -5.2942 -5.3123 -5.3304 -5.3485 -5.3666 -5.3847 -5.4028 -5.4209 -5.4390 -5.4571 -5.4752 -5.4933 -5.5114 -5.5295 -5.5476 -5.5657 -5.5838 -5.6019 -5.6200 -5.6381 -5.6562 -5.6743 -5.6924 -5.7105 -5.7286 -5.7467 -5.7648 -5.7829 -5.8010 -5.8191 -5.8372 -5.8553 -5.8734 -5.8915 -5.9096 -5.9277 -5.9458 -5.9639 -5.9820 -6.0001 -6.0182 -6.0363 -6.0544 -6.0725 -6.0906 -6.1087 -6.1268 -6.1449 -6.1630 -6.1811 -6.1992 -6.2173 -6.2354 -6.2535 -6.2716 -6.2897 -6.3078 -6.3259 -6.3440 -6.3621 -6.3802 -6.3983 -6.4164 -6.4345 -6.4526 -6.4707 -6.4888 -6.5069 -6.5250 -6.5431 -6.5612 -6.5793 -6.5974 -6.6155 -6.6336 -6.6517 -6.6698 -6.6879 -6.7060 -6.7241 -6.7422 -6.7603 -6.7784 -6.7965 -6.8146 -6.8327 -6.8508 -6.8689 -6.8870 -6.9051 -6.9232 -6.9413 -6.9594 -6.9775 -6.9956 -7.0137 -7.0318 -7.0499 -7.0680 -7.0861 -7.1042 -7.1223 -7.1404 -7.1585 -7.1766 -7.1947 -7.2128 -7.2309 -7.2490 -7.2671 -7.2852 -7.3033 -7.3214 -7.3395 -7.3576 -7.3757 -7.3938 -7.4119 -7.4300 -7.4481 -7.4662 -7.4843 -7.5024 -7.5205 -7.5386 -7.5567 -7.5748 -7.5929 -7.6110 -7.6291 -7.6472 -7.6653 -7.6834 -7.7015 -7.7196 -7.7377 -7.7558 -7.7739 -7.7920 -7.8101 -7.8282 -7.8463 -7.8644 -7.8825 -7.9006 -7.9187 -7.9368 -7.9549 -7.9730 -7.9911 -8.0092 -8.0273 -8.0454 -8.0635 -8.0816 -8.0997 -8.1178 -8.1359 -8.1540 -8.1721 -8.1902 -8.2083 -8.2264 -8.2445 -8.2626 -8.2807 -8.2988 -8.3169 -8.3350 -8.3531 -8.3712 -8.3893 -8.4074 -8.4255 -8.4436 -8.4617 -8.4798 -8.4979 -8.5160 -8.5341 -8.5522 -8.5703 -8.5884 -8.6065 -8.6246 -8.6427 -8.6608 -8.6789 -8.6970 -8.7151 -8.7332 -8.7513 -8.7694 -8.7875 -8.8056 -8.8237 -8.8418 -8.8599 -8.8780 -8.8961 -8.9142 -8.9323 -8.9504 -8.9685 -8.9866 -9.0047 -9.0228 -9.0409 -9.0590 -9.0771 -9.0952 -9.1133 -9.1314 -9.1495 -9.1676 -9.1857 -9.2038 -9.2219 -9.2400 -9.2581 -9.2762 -9.2943 -9.3124 -9.3305 -9.3486 -9.3667 -9.3848 -9.4029 -9.4210 -9.4391 -9.4572 -9.4753 -9.4934 -9.5115 -9.5296 -9.5477 -9.5658 -9.5839 -9.6020 -9.6201 -9.6382 -9.6563 -9.6744 -9.6925 -9.7106 -9.7287 -9.7468 -9.7649 -9.7830 -9.8011 -9.8192 -9.8373 -9.8554 -9.8735 -9.8916 -9.9097 -9.9278 -9.9459 -9.9640 -9.9821 -10.0000]  
    t: [5.0000e-07 1.5000e-06 2.5000e-06 3.5000e-06 4.5000e-06 5.5000e-06 6.5000e-06 7.5000e-06 8.5000e-06 9.5000e-06 1.0500e-05 1.1500e-05 1.2500e-05 1.3500e-05 1.4500e-05 1.5500e-05 1.6500e-05 1.7500e-05 1.8500e-05 1.9500e-05 2.0500e-05 2.1500e-05 2.2500e-05 2.3500e-05 2.4500e-05 2.5500e-05 2.6500e-05 2.7500e-05 2.8500e-05 2.9500e-05 3.0500e-05 3.1500e-05 3.2500e-05 3.3500e-05 3.4500e-05 3.5500e-05 3.6500e-05 3.7500e-05 3.8500e-05 3.9500e-05 4.0500e-05 4.1500e-05 4.2500e-05 4.3500e-05 4.4500e-05 4.5500e-05 4.6500e-05 4.7500e-05 4.8500e-05 4.9500e-05 5.0500e-05 5.1500e-05 5.2500e-05 5.3500e-05 5.4500e-05 5.5500e-05 5.6500e-05 5.7500e-05 5.8500e-05 5.9500e-05 6.0500e-05 6.1500e-05 6.2500e-05 6.3500e-05 6.4500e-05 6.5500e-05 6.6500e-05 6.7500e-05 6.8500e-05 6.9500e-05 7.0500e-05 7.1500e-05 7.2500e-05 7.3500e-05 7.4500e-05 7.5500e-05 7.6500e-05 7.7500e-05 7.8500e-05 7.9500e-05 8.0500e-05 8.1500e-05 8.2500e-05 8.3500e-05 8.4500e-05 8.5500e-05 8.6500e-05 8.7500e-05 8.8500e-05 8.9500e-05 9.0500e-05 9.1500e-05 9.2500e-05 9.3500e-05 9.4500e-05 9.5500e-05 9.6500e-05 9.7500e-05 9.8500e-05 9.9500e-05 1.0000e-04 1.0100e-04 1.0200e-04 1.0300e-04 1.0400e-04 1.0500e-04 1.0600e-04 1.0700e-04 1.0800e-04 1.0900e-04 1.1000e-04 1.1100e-04 1.1200e-04 1.1300e-04 1.1400e-04 1.1500e-04 1.1600e-04 1.1700e-04 1.1800e-04 1.1900e-04 1.2000e-04 1.2100e-04 1.2200e-04 1.2300e-04 1.2400e-04 1.2500e-04 1.2600e-04 1.2700e-04 1.2800e-04 1.2900e-04 1.3000e-04 1.3100e-04 1.3200e-04 1.3300e-04 1.3400e-04 1.3500e-04 1.3600e-04 1.3700e-04 1.3800e-04 1.3900e-04 1.4000e-04 1.4100e-04 1.4200e-04 1.4300e-04 1.4400e-04 1.4500e-04 1.4600e-04 1.4700e-04 1.4800e-04 1.4900e-04 1.5000e-04 1.5100e-04 1.5200e-04 1.5300e-04 1.5400e-04 1.5500e-04 1.5600e-04 1.5700e-04 1.5800e-04 1.5900e-04 1.6000e-04 1.6100e-04 1.6200e-04 1.6300e-04 1.6400e-04 1.6500e-04 1.6600e-04 1.6700e-04 1.6800e-04 1.6900e-04 1.7000e-04 1.7100e-04 1.7200e-04 1.7300e-04 1.7400e-04 1.7500e-04 1.7600e-04 1.7700e-04 1.7800e-04 1.7900e-04 1.8000e-04 1.8100e-04 1.8200e-04 1.8300e-04 1.8400e-04 1.8500e-04 1.8600e-04 1.8700e-04 1.8800e-04 1.8900e-04 1.9000e-04 1.9100e-04 1.9200e-04 1.9300e-04 1.9400e-04 1.9500e-04 1.9600e-04 1.9700e-04 1.9800e-04 1.9900e-04 2.0000e-04 2.0100e-04 2.0200e-04 2.0300e-04 2.0400e-04 2.0500e-04 2.0600e-04 2.0700e-04 2.0800e-04 2.0900e-04 2.1000e-04 2.1100e-04 2.1200e-04 2.1300e-04 2.1400e-04 2.1500e-04 2.1600e-04 2.1700e-04 2.1800e-04 2.1900e-04 2.2000e-04 2.2100e-04 2.2200e-04 2.2300e-04 2.2400e-04 2.2500e-04 2.2600e-04 2.2700e-04 2.2800e-04 2.2900e-04 2.3000e-04 2.3100e-04 2.3200e-04 2.3300e-04 2.3400e-04 2.3500e-04 2.3600e-04 2.3700e-04 2.3800e-04 2.3900e-04 2.4000e-04 2.4100e-04 2.4200e-04 2.4300e-04 2.4400e-04 2.4500e-04 2.4600e-04 2.4700e-04 2.4800e-04 2.4900e-04 2.5000e-04 2.5100e-04 2.5200e-04 2.5300e-04 2.5400e-04 2.5500e-04 2.5600e-04 2.5700e-04 2.5800e-04 2.5900e-04 2.6000e-04 2.6100e-04 2.6200e-04 2.6300e-04 2.6400e-04 2.6500e-04 2.6600e-04 2.6700e-04 2.6800e-04 2.6900e-04 2.7000e-04 2.7100e-04 2.7200e-04 2.7300e-04 2.7400e-04 2.7500e-04 2.7600e-04 2.7700e-04 2.7800e-04 2.7900e-04 2.8000e-04 2.8100e-04 2.8200e-04 2.8300e-04 2.8400e-04 2.8500e-04 2.8600e-04 2.8700e-04 2.8800e-04 2.8900e-04 2.9000e-04 2.9100e-04 2.9200e-04 2.9300e-04 2.9400e-04 2.9500e-04 2.9600e-04 2.9700e-04 2.9800e-04 2.9900e-04 3.0000e-04 3.0100e-04 3.0200e-04 3.0300e-04 3.0400e-04 3.0500e-04 3.0600e-04 3.0700e-04 3.0800e-04 3.0900e-04 3.1000e-04 3.1100e-04 3.1200e-04 3.1300e-04 3.1400e-04 3.1500e-04 3.1600e-04 3.1700e-04 3.1800e-04 3.1900e-04 3.2000e-04 3.2100e-04 3.2200e-04 3.2300e-04 3.2400e-04 3.2500e-04 3.2600e-04 3.2700e-04 3.2800e-04 3.2900e-04 3.3000e-04 3.3100e-04 3.3200e-04 3.3300e-04 3.3400e-04 3.3500e-04 3.3600e-04 3.3700e-04 3.3800e-04 3.3900e-04 3.4000e-04 3.4100e-04 3.4200e-04 3.4300e-04 3.4400e-04 3.4500e-04 3.4600e-04 3.4700e-04 3.4800e-04 3.4900e-04 3.5000e-04 3.5100e-04 3.5200e-04 3.5300e-04 3.5400e-04 3.5500e-04 3.5600e-04 3.5700e-04 3.5800e-04 3.5900e-04 3.6000e-04 3.6100e-04 3.6200e-04 3.6300e-04 3.6400e-04 3.6500e-04 3.6600e-04 3.6700e-04 3.6800e-04 3.6900e-04 3.7000e-04 3.7100e-04 3.7200e-04 3.7300e-04 3.7400e-04 3.7500e-04 3.7600e-04 3.7700e-04 3.7800e-04 3.7900e-04 3.8000e-04 3.8100e-04 3.8200e-04 3.8300e-04 3.8400e-04 3.8500e-04 3.8600e-04 3.8700e-04 3.8800e-04 3.8900e-04 3.9000e-04 3.9100e-04 3.9200e-04 3.9300e-04 3.9400e-04 3.9500e-04 3.9600e-04 3.9700e-04 3.9800e-04 3.9900e-04 4.0000e-04 4.0100e-04 4.0200e-04 4.0300e-04 4.0400e-04 4.0500e-04 4.0600e-04 4.0700e-04 4.0800e-04 4.0900e-04 4.1000e-04 4.1100e-04 4.1200e-04 4.1300e-04 4.1400e-04 4.1500e-04 4.1600e-04 4.1700e-04 4.1800e-04 4.1900e-04 4.2000e-04 4.2100e-04 4.2200e-04 4.2300e-04 4.2400e-04 4.2500e-04 4.2600e-04 4.2700e-04 4.2800e-04 4.2900e-04 4.3000e-04 4.3100e-04 4.3200e-04 4.3300e-04 4.3400e-04 4.3500e-04 4.3600e-04 4.3700e-04 4.3800e-04 4.3900e-04 4.4000e-04 4.4100e-04 4.4200e-04 4.4300e-04 4.4400e-04 4.4500e-04 4.4600e-04 4.4700e-04 4.4800e-04 4.4900e-04 4.5000e-04 4.5100e-04 4.5200e-04 4.5300e-04 4.5400e-04 4.5500e-04 4.5600e-04 4.5700e-04 4.5800e-04 4.5900e-04 4.6000e-04 4.6100e-04 4.6200e-04 4.6300e-04 4.6400e-04 4.6500e-04 4.6600e-04 4.6700e-04 4.6800e-04 4.6900e-04 4.7000e-04 4.7100e-04 4.7200e-04 4.7300e-04 4.7400e-04 4.7500e-04 4.7600e-04 4.7700e-04 4.7800e-04 4.7900e-04 4.8000e-04 4.8100e-04 4.8200e-04 4.8300e-04 4.8400e-04 4.8500e-04 4.8600e-04 4.8700e-04 4.8800e-04 4.8900e-04 4.9000e-04 4.9100e-04 4.9200e-04 4.9300e-04 4.9400e-04 4.9500e-04 4.9600e-04 4.9700e-04 4.9800e-04 4.9900e-04 5.0000e-04 5.0100e-04 5.0200e-04 5.0300e-04 5.0400e-04 5.0500e-04 5.0600e-04 5.0700e-04 5.0800e-04 5.0900e-04 5.1000e-04 5.1100e-04 5.1200e-04 5.1300e-04 5.1400e-04 5.1500e-04 5.1600e-04 5.1700e-04 5.1800e-04 5.1900e-04 5.2000e-04 5.2100e-04 5.2200e-04 5.2300e-04 5.2400e-04 5.2500e-04 5.2600e-04 5.2700e-04 5.2800e-04 5.2900e-04 5.3000e-04 5.3100e-04 5.3200e-04 5.3300e-04 5.3400e-04 5.3500e-04 5.3600e-04 5.3700e-04 5.3800e-04 5.3900e-04 5.4000e-04 5.4100e-04 5.4200e-04 5.4300e-04 5.4400e-04 5.4500e-04 5.4600e-04 5.4700e-04 5.4800e-04 5.4900e-04 5.5000e-04 5.5100e-04 5.5200e-04 5.5300e-04 5.5400e-04 5.5500e-04 5.5600e-04 5.5700e-04 5.5800e-04 5.5900e-04 5.6000e-04 5.6100e-04 5.6200e-04 5.6300e-04 5.6400e-04 5.6500e-04 5.6600e-04 5.6700e-04 5.6800e-04 5.6900e-04 5.7000e-04 5.7100e-04 5.7200e-04 5.7300e-04 5.7400e-04 5.7500e-04 5.7600e-04 5.7700e-04 5.7800e-04 5.7900e-04 5.8000e-04 5.8100e-04 5.8200e-04 5.8300e-04 5.8400e-04 5.8500e-04 5.8600e-04 5.8700e-04 5.8800e-04 5.8900e-04 5.9000e-04 5.9100e-04 5.9200e-04 5.9300e-04 5.9400e-04 5.9500e-04 5.9600e-04 5.9700e-04 5.9800e-04 5.9900e-04 6.0000e-04 6.0100e-04 6.0200e-04 6.0300e-04 6.0400e-04 6.0500e-04 6.0600e-04 6.0700e-04 6.0800e-04 6.0900e-04 6.1000e-04 6.1100e-04 6.1200e-04 6.1300e-04 6.1400e-04 6.1500e-04 6.1600e-04 6.1700e-04 6.1800e-04 6.1900e-04 6.2000e-04 6.2100e-04 6.2200e-04 6.2300e-04 6.2400e-04 6.2500e-04 6.2600e-04 6.2700e-04 6.2800e-04 6.2900e-04 6.3000e-04 6.3100e-04 6.3200e-04 6.3300e-04 6.3400e-04 6.3500e-04 6.3600e-04 6.3700e-04 6.3800e-04 6.3900e-04 6.4000e-04 6.4100e-04 6.4200e-04 6.4300e-04 6.4400e-04 6.4500e-04 6.4600e-04 6.4700e-04 6.4800e-04 6.4900e-04 6.5000e-04 6.5100e-04 6.5200e-04 6.5300e-04 6.5400e-04 6.5500e-04 6.5600e-04 6.5700e-04 6.5800e-04 6.5900e-04 6.6000e-04 6.6100e-04 6.6200e-04 6.6300e-04 6.6400e-04 6.6500e-04 6.6600e-04 6.6700e-04 6.6800e-04 6.6900e-04 6.7000e-04 6.7100e-04 6.7200e-04 6.7300e-04 6.7400e-04 6.7500e-04 6.7600e-04 6.7700e-04 6.7800e-04 6.7900e-04 6.8000e-04 6.8100e-04 6.8200e-04 6.8300e-04 6.8400e-04 6.8500e-04 6.8600e-04 6.8700e-04 6.8800e-04 6.8900e-04 6.9000e-04 6.9100e-04 6.9200e-04 6.9300e-04 6.9400e-04 6.9500e-04 6.9600e-04 6.9700e-04 6.9800e-04 6.9900e-04 7.0000e-04 7.0100e-04 7.0200e-04 7.0300e-04 7.0400e-04 7.0500e-04 7.0600e-04 7.0700e-04 7.0800e-04 7.0900e-04 7.1000e-04 7.1100e-04 7.1200e-04 7.1300e-04 7.1400e-04 7.1500e-04 7.1600e-04 7.1700e-04 7.1800e-04 7.1900e-04 7.2000e-04 7.2100e-04 7.2200e-04 7.2300e-04 7.2400e-04 7.2500e-04 7.2600e-04 7.2700e-04 7.2800e-04 7.2900e-04 7.3000e-04 7.3100e-04 7.3200e-04 7.3300e-04 7.3400e-04 7.3500e-04 7.3600e-04 7.3700e-04 7.3800e-04 7.3900e-04 7.4000e-04 7.4100e-04 7.4200e-04 7.4300e-04 7.4400e-04 7.4500e-04 7.4600e-04 7.4700e-04 7.4800e-04 7.4900e-04 7.5000e-04 7.5100e-04 7.5200e-04 7.5300e-04 7.5400e-04 7.5500e-04 7.5600e-04 7.5700e-04 7.5800e-04 7.5900e-04 7.6000e-04 7.61
```

# Creating blocks

```
seq = mr.Sequence();

% slice-select pulse
sincrf;
seq.addBlock(rf, gz, mr.makeDelay(2e-3)); % create rf and gz
                                           % what is effect of the delay event here??

% slice rephaser gradient
gzrep = mr.makeTrapezoid('z', 'Area', -gz.area/2, 'system', sys);
seq.addBlock(gzrep);

% spiral readout
spiral; % create gx
seq.addBlock(gx, adc);

close all; seq.plot('showBlocks', true);
```

```
seq.getBlock(1) =
    blockDuration: 0.0022
         rf: [1x1 struct]
         gx: []
         gy: []
         gz: [1x1 struct]
         adc: []
```

# Creating blocks

```
seq = mr.Sequence();

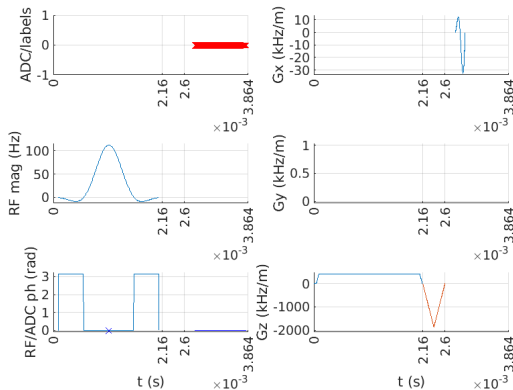
% slice-select pulse
sincrf;
seq.addBlock(rf, gz, mr.makeDelay(2e-3)); % create rf and gz
                                           % what is effect of the delay event here??

% slice rephaser gradient
gzrep = mr.makeTrapezoid('z', 'Are);
seq.addBlock(gzrep);

% spiral readout
spiral; % create spiral
seq.addBlock(gx, adc);

close all; seq.plot('showBlocks',
```

```
seq.getBlock(1) =
    blockDuration: 0.0022
         rf: [1x1 struct]
         gx: []
         gy: []
         gz: [1x1 struct]
        adc: []
```



# Creating blocks

## Rules:

- Time is referenced to start of block
- Block duration is determined by the longest event
- Only one event on each channel (per block)

To pass `seq.checkTiming()`, additional rules apply:

- Minimum RF delay = `sys.rfDeadTime`
- Minimum RF ringdown time before end of block = `sys.rfRingdownTime`
- Minimum ADC delay = `sys.adcDeadTime`
- Minimum time after ADC event before end of block = `sys.adcDeadTime`
- Many more constraints are verified/checked by `checkTiming()`, which will be covered in the 'Sequence Analysis' talk.