

# 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

# .seq file structure

```
# Pulseseq sequence file , truncated for illustration purposes

# NUM DUR RF  GX  GY  GZ  ADC  EXT
[BLOCKS]
  1 320   1   0   0   1   0   1
  2 100   0   2   3   4   0   0

# id amplitude mag_id phase_id time_shape_id delay freq phase
[RF]
1      22.3311 1 2 0 100 0 0
2      22.3311 1 2 0 100 0 2.04204

# id amplitude rise flat fall delay
[TRAP]
1      266667 100 3000 100 0
2     -150709 60 880 60 0

# id num dwell delay freq phase
[ADC]
1 32 10000 190 0 0
2 32 10000 190 0 2.04204

# Sequence Shapes
[SHAPES]

shape_id 1
num_samples 3000
5.33512061e-05
0.000160161335
0.000267116839
```

# Gradient events

3 types:

- Trapezoid
- Extended trapezoid
- Arbitrary gradient

Pulseseq toolbox defaults

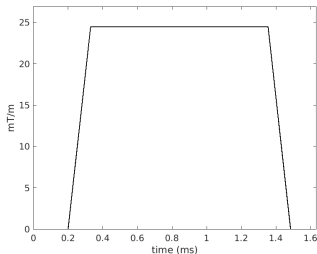
- `gradRasterTime` = 10e-6 s

# 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
```



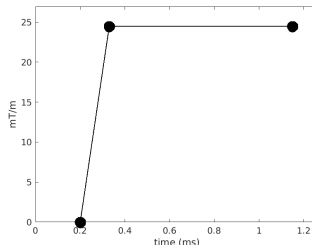
# Extended trapezoid

## Shape sampled on vertices

```
gxtmp = mr.makeTrapezoid('x', 'FlatArea', Nx*deltak, 'FlatTime', Nx*dwel, ...  
                          '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);
```

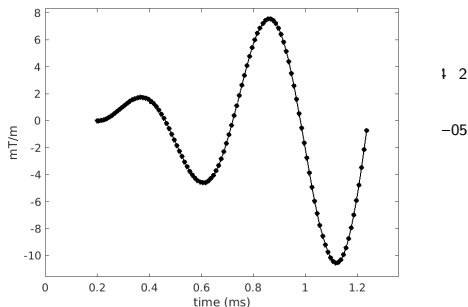
```
gx =
  type: 'grad'
  channel: 'x'
  waveform: [148.3082 1.3283e+03 3.6538e+03 7.0567e+03 1.1437e+04 1.6663e+04 2.2579e+04 2.
  delay: 2.0000e-04
  area: -42.3091
  tt: [5.0000e-06 1.5000e-05 2.5000e-05 3.5000e-05 4.5000e-05 5.5000e-05 6.5000e-05
  shape_dur: 0.0010
  first: -441.6805
  last: -666.5979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

# 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);
```

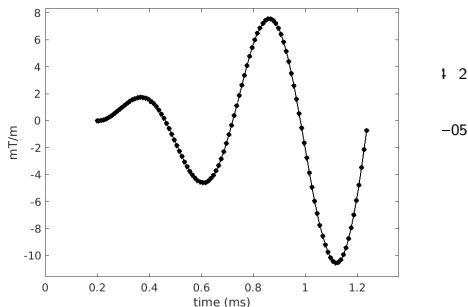
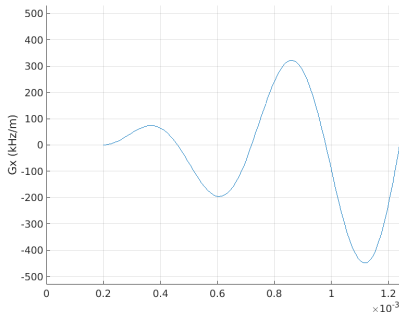
```
gx =  
    type: 'grad'  
    channel: 'x'  
    waveform: [148.3082 1.3283e+03 3.6538e+03]  
    delay: 2.0000e-04  
    area: -42.3091  
    tt: [5.0000e-06 1.5000e-05 2.5000e-05]  
    shape_dur: 0.0010  
    first: -441.6805  
    last: -666.5979
```



# Arbitrary gradient

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

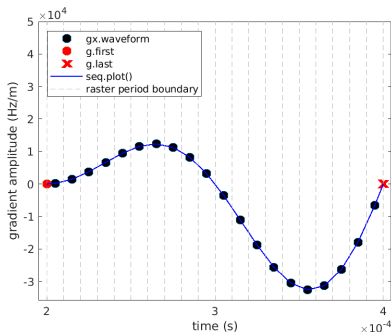
```
spiral; % creates 'wav' vector, mT/m  
gx = mr.makeArbitraryGrad('x', wav*sys.gamma/1e3, 'delay', 200e-6);
```



## '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
```

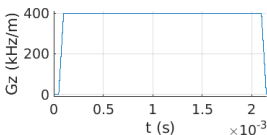
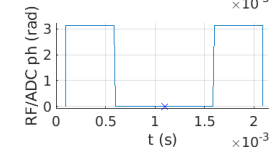
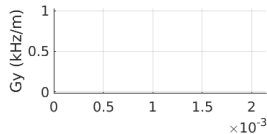
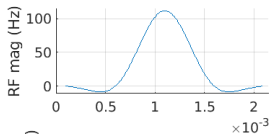
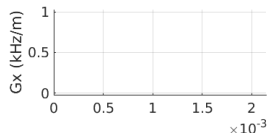
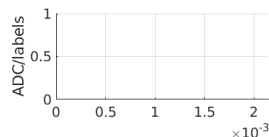
# RF event: Example 1

## 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 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;
```

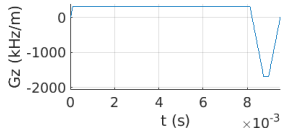
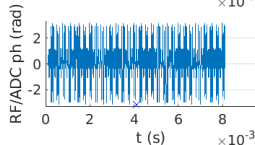
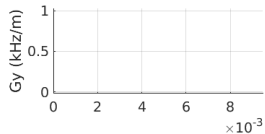
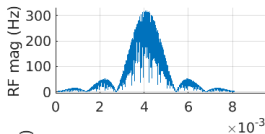
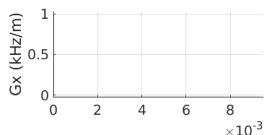
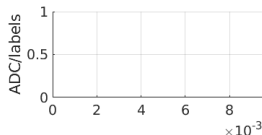
```
gz = mr.makeArbitraryGrad('z', g
```

```
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 -1.9993 -2.0174 -2.0355 -2.0536 -2.0717 -2.0898 -2.1079 -2.1260 -2.1441 -2.1622 -2.1803 -2.1984 -2.2165 -2.2346 -2.2527 -2.2708 -2.2889 -2.3070 -2.3251 -2.3432 -2.3613 -2.3794 -2.3975 -2.4156 -2.4337 -2.4518 -2.4699 -2.4880 -2.5061 -2.5242 -2.5423 -2.5604 -2.5785 -2.5966 -2.6147 -2.6328 -2.6509 -2.6690 -2.6871 -2.7052 -2.7233 -2.7414 -2.7595 -2.7776 -2.7957 -2.8138 -2.8319 -2.8500 -2.8681 -2.8862 -2.9043 -2.9224 -2.9405 -2.9586 -2.9767 -2.9948 -3.0129 -3.0310 -3.0491 -3.0672 -3.0853 -3.1034 -3.1215 -3.1396 -3.1577 -3.1758 -3.1939 -3.2120 -3.2301 -3.2482 -3.2663 -3.2844 -3.3025 -3.3206 -3.3387 -3.3568 -3.3749 -3.3930 -3.4111 -3.4292 -3.4473 -3.4654 -3.4835 -3.5016 -3.5197 -3.5378 -3.5559 -3.5740 -3.5921 -3.6102 -3.6283 -3.6464 -3.6645 -3.6826 -3.7007 -3.7188 -3.7369 -3.7550 -3.7731 -3.7912 -3.8093 -3.8274 -3.8455 -3.8636 -3.8817 -3.9000 -3.9181 -3.9362 -3.9543 -3.9724 -3.9905 -4.0086 -4.0267 -4.0448 -4.0629 -4.0810 -4.0991 -4.1172 -4.1353 -4.1534 -4.1715 -4.1896 -4.2077 -4.2258 -4.2439 -4.2620 -4.2801 -4.2982 -4.3163 -4.3344 -4.3525 -4.3706 -4.3887 -4.4068 -4.4249 -4.4430 -4.4611 -4.4792 -4.4973 -4.5154 -4.5335 -4.5516 -4.5697 -4.5878 -4.6059 -4.6240 -4.6421 -4.6602 -4.6783 -4.6964 -4.7145 -4.7326 -4.7507 -4.7688 -4.7869 -4.8050 -4.8231 -4.8412 -4.8593 -4.8774 -4.8955 -4.9136 -4.9317 -4.9498 -4.9679 -4.9860 -5.0041 -5.0222 -5.0403 -5.0584 -5.0765 -5.0946 -5.1127 -5.1308 -5.1489 -5.1670 -5.1851 -5.2032 -5.2213 -5.2394 -5.2575 -5.2756 -5.2937 -5.3118 -5.3299 -5.3480 -5.3661 -5.3842 -5.4023 -5.4204 -5.4385 -5.4566 -5.4747 -5.4928 -5.5109 -5.5290 -5.5471 -5.5652 -5.5833 -5.6014 -5.6195 -5.6376 -5.6557 -5.6738 -5.6919 -5.7100 -5.7281 -5.7462 -5.7643 -5.7824 -5.8005 -5.8186 -5.8367 -5.8548 -5.8729 -5.8910 -5.9091 -5.9272 -5.9453 -5.9634 -5.9815 -6.0000 -6.0181 -6.0362 -6.0543 -6.0724 -6.0905 -6.1086 -6.1267 -6.1448 -6.1629 -6.1810 -6.1991 -6.2172 -6.2353 -6.2534 -6.2715 -6.2896 -6.3077 -6.3258 -6.3439 -6.3620 -6.3801 -6.3982 -6.4163 -6.4344 -6.4525 -6.4706 -6.4887 -6.5068 -6.5249 -6.5430 -6.5611 -6.5792 -6.5973 -6.6154 -6.6335 -6.6516 -6.6697 -6.6878 -6.7059 -6.7240 -6.7421 -6.7602 -6.7783 -6.7964 -6.8145 -6.8326 -6.8507 -6.8688 -6.8869 -6.9050 -6.9231 -6.9412 -6.9593 -6.9774 -6.9955 -7.0136 -7.0317 -7.0498 -7.0679 -7.0860 -7.1041 -7.1222 -7.1403 -7.1584 -7.1765 -7.1946 -7.2127 -7.2308 -7.2489 -7.2670 -7.2851 -7.3032 -7.3213 -7.3394 -7.3575 -7.3756 -7.3937 -7.4118 -7.4299 -7.4480 -7.4661 -7.4842 -7.5023 -7.5204 -7.5385 -7.5566 -7.5747 -7.5928 -7.6109 -7.6290 -7.6471 -7.6652 -7.6833 -7.7014 -7.7195 -7.7376 -7.7557 -7.7738 -7.7919 -7.8100 -7.8281 -7.8462 -7.8643 -7.8824 -7.9005 -7.9186 -7.9367 -7.9548 -7.9729 -7.9910 -8.0091 -8.0272 -8.0453 -8.0634 -8.0815 -8.0996 -8.1177 -8.1358 -8.1539 -8.1720 -8.1901 -8.2082 -8.2263 -8.2444 -8.2625 -8.2806 -8.2987 -8.3168 -8.3349 -8.3530 -8.3711 -8.3892 -8.4073 -8.4254 -8.4435 -8.4616 -8.4797 -8.4978 -8.5159 -8.5340 -8.5521 -8.5702 -8.5883 -8.6064 -8.6245 -8.6426 -8.6607 -8.6788 -8.6969 -8.7150 -8.7331 -8.7512 -8.7693 -8.7874 -8.8055 -8.8236 -8.8417 -8.8598 -8.8779 -8.8960 -8.9141 -8.9322 -8.9503 -8.9684 -8.9865 -9.0046 -9.0227 -9.0408 -9.0589 -9.0770 -9.0951 -9.1132 -9.1313 -9.1494 -9.1675 -9.1856 -9.2037 -9.2218 -9.2399 -9.2580 -9.2761 -9.2942 -9.3123 -9.3304 -9.3485 -9.3666 -9.3847 -9.4028 -9.4209 -9.4390 -9.4571 -9.4752 -9.4933 -9.5114 -9.5295 -9.5476 -9.5657 -9.5838 -9.6019 -9.6200 -9.6381 -9.6562 -9.6743 -9.6924 -9.7105 -9.7286 -9.7467 -9.7648 -9.7829 -9.8010 -9.8191 -9.8372 -9.8553 -9.8734 -9.8915 -9.9096 -9.9277 -9.9458 -9.9639 -9.9820 -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();

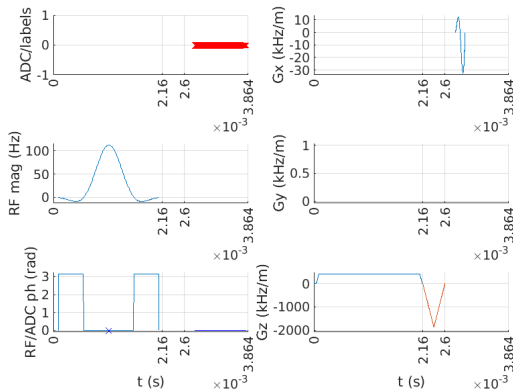
% slice-select pulse
sincrf;
seq.addBlock(rf, gz, mr.makeDelay(2e-3)); % create rf and gz
                                           % what is effect of the delay event here??

% 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.