

## Installation and user's manual for the Free Waveform Encoding (FWF) sequence (s)

### Scope

This document provides a very brief user manual to the FWF pulse sequence. For detailed information, please contact the author or find further information at the sequence resource page:

[https://github.com/filip-szczepankiewicz/fwf\\_seq\\_resources](https://github.com/filip-szczepankiewicz/fwf_seq_resources)

### Introduction

The FWF sequence is based on the diffusion-weighted spin-echo diffusion encoding sequence. It removes the trapezoidal diffusion encoding waveforms and replaces them with an arbitrary waveform. The waveforms are compiled into the sequence and cannot be modified by the user.

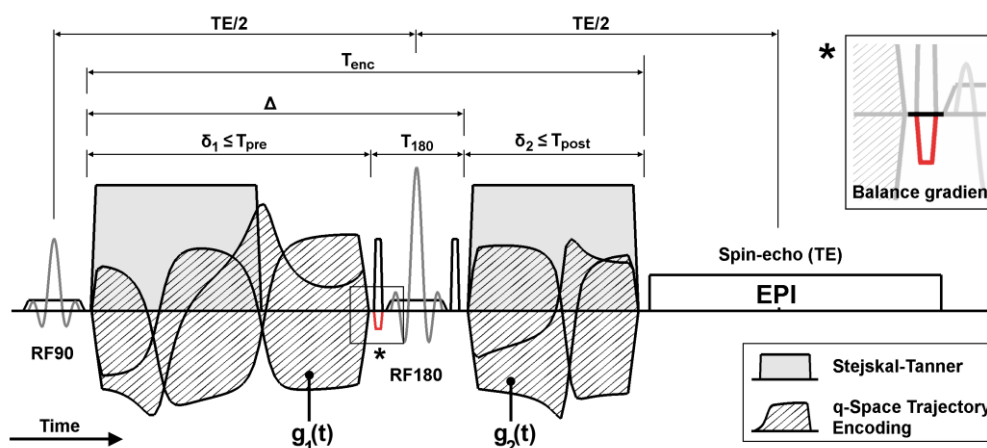


Fig. 1 – Schematic spin-echo sequence and its timing variables. The gradient waveforms are executed between the excitation pulse, refocusing pulse, and echo-planar readout. The timing variables show the maximal time available for encoding ( $T_{pre}$  and  $T_{post}$ ), the duration of each gradient waveform ( $\delta_1$  and  $\delta_2$ ), the gradient waveform separation ( $\Delta$ ), the duration of the refocusing block including the crushers ( $T_{180}$ ), the total encoding time ( $T_{enc}$ ), and the echo time ( $TE$ ). The balance gradient (red) is executed at the same time as the first crusher. Note that the Stejskal-Tanner and q-space trajectory encoding gradients are shown together for visual reference but are not executed simultaneously in practice. Image adapted from [1].

### Installation of sequence

1. Transfer the provided .dll and .so files to the customer sequence folder. This folder can be reached by writing %CustomerSeq% in the explorer address window, and pressing enter, and may have a different absolute file path on different systems.
2. Import the sequence into the DOT editor (main interface on host computer) by dragging it from the default customer sequence “folder”. Please see the separate manual for this step.

### First use of FWF sequence

**Note: Most situations require that the sequence be set up from the default settings, as opposed to importing settings from a DICOM or exar1-files!**

1. Load the sequence into the sequence editor. Check that the loaded sequence states that ‘a\_ep2d\_diff\_fwf\_simple’ is running by hovering the cursor over the sequence type (normally seen as the string ‘epse’ in the upper right corner, see green square in Figure 2).

2. To engage the additional FWF functions, select the [Diff]-tab, and set the 'Diffusion Scheme' to 'Monopolar', instead of 'Bipolar' (see red square in Figure 2). When the 'Monopolar' encoding type is selected, the FWF sequence is engaged, and it replaces the default monopolar Stejskal-Tanner sequence. To run the original monopolar version, please use the standard sequence from the sequence library.
3. Go to [Sequence] – [Special] – tab. All FWF functionality is controlled from this tab. Conventional controls, such as setting the b-values, are unchanged. The special tab should look as that presented in Figure 1. If the parameter 'MaxBVal' is zero, something is wrong, and the sequence should not be executed!
4. Change the default b-value (default is zero, see purple box in Figure 2) and test the sequence on a water phantom.
  - a. It is a good idea to start slow, and to validate that the sequence measures the correct ADC in water.
  - b. Also test the sequence in oil and note if any signal is lost (it should not be unless you go to very high b-values [2]).

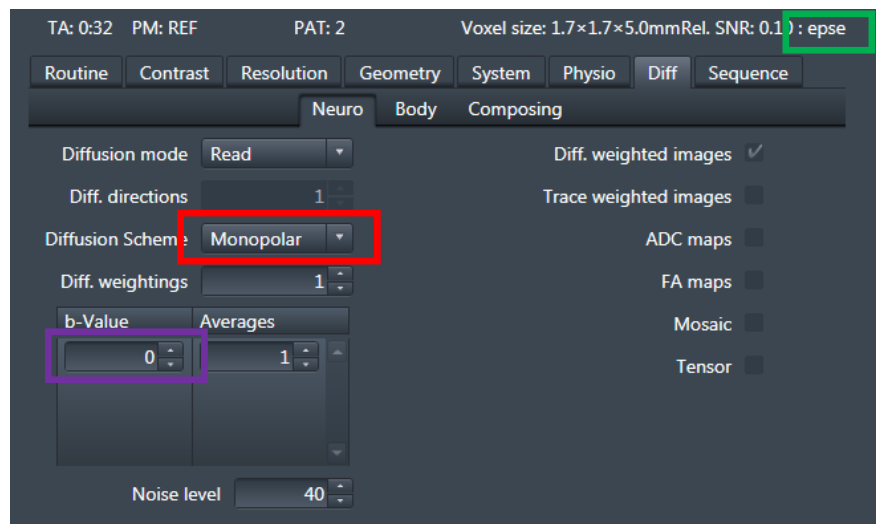


Fig. 2 – Select the “monopolar” diffusion scheme to engage the FWF sequence (red box). Set the b-value using the same method as for the default sequence (purple box).

### The [Sequence] – [Special]-tab

All new functions are modified in the [Sequence] – [Special]-tab (Figure 3); standard functions are unaltered. NOTE: additional information is available in the tool-tips that appear when hovering over the parameter selectors. Figure 3 shows an example of the interface that is used to control the FWF functions. Hover over the 'MaxBVal' value to get comprehensive sequence info, and hover over either of the 'Dur' values to see the timing of the sequence.

In the “simple” variant of the FWF sequence, the only parameter that can be changed is the shape of the b-tensor. The drop-down menu allows for selection of linear, planar, and spherical encoding (LTE, PTE, STE). Variants of the waveforms may be present for specific applications. Note that these are not equally efficient and may require that other parameters be set based on the least efficient waveform (usually STE).

## Hints

1. To get some useful information about the sequence, its timing, and the hardware specifications, you can hover the cursor over the parameters in the special tab.
2. Do not run the waveforms at a maximal gradient amplitude (max possible b-value or minimal echo time). Try to leave approximately 5 mT/m headroom. This can be done by adding a few milliseconds to the minimal echo time.
3. Do not set the TE or TR at their minimal values since future software updates may change these slightly and force a change in an ongoing study.
4. Comprehensive information about the sequence is available in the real-time logging of the system. In the log-viewer, search for the keyword “galore”, to find the FWF-specific info.
5. Start by setting up the most challenging waveform (the one that yields the lowest b-value for a given setup) and use identical imaging parameters for all the other exam cards and b-tensor shapes. In other words, each examcard should only differ in what waveforms is used and what sampling scheme (dvs-file) is executed!



Fig. 3 – Example of the [Special]-tab in the simple sequence. When the sequence is set to ‘Monopolar’, this tab will be active and updated.

## Example setup

Setup the imaging protocol (these are a good starting point for whole-brain imaging).

- FOV = 220x220x120 mm<sup>3</sup>
- Matrix = 110x110
- 30-50 Slices
- Resolution
  - 2x2x4 mm<sup>3</sup>
  - 2.3x2.3x2.3 mm<sup>3</sup>
- TE ≈ 85
  - Minimal value + 4 ms to create headroom
- Partial Fourier = 6/8
- iPAT = 2 (GRAPPA)
- Bandwidth 1800 Hz/pix

- $b = .1, .7, 1.4, \text{ and } 2 \text{ ms}/\mu\text{m}^2$ 
  - Avoid using  $b = 0$  images in analysis
  - See reference for data quality comparison [3]
- Num. diff. dirs = 6, 6, 12, 16 for linear encoding.
  - The same number of samples can be used for PTE or STE too
  - More directions can be added, starting from the outer shells
  - See resources below for sampling schemes

## Resources

Main resource page contains waveform definitions, sampling schemes and links to other useful tools:  
[https://github.com/filip-szczepankiewicz/fwf\\_seq\\_resources](https://github.com/filip-szczepankiewicz/fwf_seq_resources)

Overview of design considerations when using free waveforms and b-tensor encoding [4]:  
<https://www.sciencedirect.com/science/article/pii/S0165027020304301>

Example protocols and sampling schemes from ref [3]:  
[https://github.com/filip-szczepankiewicz/Szczepankiewicz\\_PONE\\_2019](https://github.com/filip-szczepankiewicz/Szczepankiewicz_PONE_2019)

Analysis software [5]:  
<https://github.com/markus-nilsson/md-dmri>

Extraction of FWF-specific header information from DICOM images:  
[https://github.com/filip-szczepankiewicz/fwf\\_header\\_tools](https://github.com/filip-szczepankiewicz/fwf_header_tools)

## References

1. Szczepankiewicz, F., *Imaging diffusional variance by MRI: The role of tensor-valued diffusion encoding and tissue heterogeneity*, in *Department of Medical Radiation Physics*. 2016, Lund University.
2. Szczepankiewicz, F., C.F. Westin, and M. Nilsson, *Maxwell-compensated design of asymmetric gradient waveforms for tensor-valued diffusion encoding*. *Magn Reson Med*, 2019.
3. Szczepankiewicz, F., et al., *Tensor-valued diffusion encoding for diffusional variance decomposition (DIVIDE): Technical feasibility in clinical MRI systems*. *PLoS One*, 2019. **14**(3): p. e0214238.
4. Szczepankiewicz, F., C.F. Westin, and M. Nilsson, *Gradient waveform design for tensor-valued encoding in diffusion MRI*. *J Neurosci Methods*, 2021: p. 109007.
5. Nilsson, M., et al. *An open-source framework for analysis of multidimensional diffusion MRI data implemented in MATLAB*. in *Proc. Intl. Soc. Mag. Reson. Med.* 26. 2018. Paris, France.