

User Guide for MATLAB

Extended Phase Graph (EPG) simulation scripts

Author:   
Gehua Tong

Graduate Research Assistant

Columbia MR Research Program (CMRRP),

School of Engineering and Applied Sciences (SEAS),

Biomedical Engineering Department

Columbia University

Email: gt2407@columbia.edu

**Table of Contents**

[1. Introduction 3](#_Toc504147383)

[2. Using the EPG functions 4](#_Toc504147384)

[2.1 Operators 4](#_Toc504147385)

[2.2 Custom EPG 6](#_Toc504147386)

[2.2 Pre-defined pulse sequences 6](#_Toc504147386)

[2.3 Displaying EPGs and echoes 6](#_Toc504147387)

[3. References 8](#_Toc504147389)

# 1. Introduction

This document describes how to use the EPG scripts for multi-echo MR pulse sequence simulation. The inputs and outputs of functions are specified, and a detailed demo is included for the Turbo Spin Echo sequence.

The scripts have been developed by Gehua Tong and Sairam Geethanath based on the 2015 paper:

Weigel M. Extended phase graphs: dephasing, RF pulses, and echoes‐pure and simple. Journal of Magnetic Resonance Imaging. 2015 Feb;41(2):266-95.

The extended phase graph algorithm is an alternative to time-domain Bloch simulations. Operating in the Fourier domain, it enables straightforward echo detection and is valuable for multi-echo sequences such as Turbo Spin Echo. The current scripts are capable of simulating the response to regularly spaced RF pulses with arbitrary phase and flip angle with linear gradients and T1, T2 relaxation effects. By “regularly spaced”, we mean that the intervals between pulses are integer multiples of the same Δt. Arbitrary spacing can be approximated with a small Δt and a large number of intervals, but the simulation will take a longer time.

You can familiarize yourself with EPG math by reading the paper cited above, or by reading the power point slides included in the EPG package.

# 

# 2. Using the EPG functions

EPG simulation can be accomplished by simply installing MATLAB (R2018a) and running the functions. A list of core functions is provided below:

* rf\_rotation.m
* shift\_grad.m
* relax.m
* EPG\_custom.m
* findEchoes.m
* display\_epg.m

Other scripts named EPGsim\_X.m are predefined pulse sequences that take inputs of parameters and use these core functions to simulate EPGs. Their usage is straightforward if you understand the core functions. Some examples are included in Section 2.3.

## 2.1 Operators

EPG represents the magnetization in terms of configuration states. At any time, the configuration matrix Ω looks like this:

There are three basic operators in EPG simulation: RF pulse, gradient shift, and T1 & T2 relaxation, represented by the functions **rf\_rotation, shift\_grad,** and **relax**.

* rf\_rotation(phi,alpha) returns a 3 x 3 matrix
* shift\_grad(dk,omega) takes the configuration states and shifts it by an integer dk
* relax(tau,T1,T2,omega) returns the new omega after relaxation over interval tau

2.1.1 RF Rotation

The rotation operator exchanges the F+, F-, and Z populations within each k.

The operator is a matrix multiplication: where

2.1.2 Gradient shift

A positive unit gradient (dk = 1) moves all the F populations to a higher k but keeps the Z populations in place. For example:

With a negative gradient, the populations move in the opposite direction, and as gradients are applied, the matrix grows more columns.

2.1.3 T1 & T2 Relaxation

T1 and T2 relaxation are represented together in the following operator. Over an interval τ, the relaxation factors are .

For k = 0,

And for k ≠0,

## 2.2 Custom EPG

Use EPG\_custom.m to simulate your own pulse sequence. The struct seq needs to have the appropriate fields defined. Here is an example:

seq.rf = [ 0 0 0

90 180 180];

seq.grad =[1,1,1,1];

seq.events = {'rf','grad','relax','rf','grad','relax','grad','relax','rf','grad','relax'};

seq.time = [0,10,10,10,20,20,30,30,30,40,40,];

seq.T1 = 1000; seq.T2 = 100;

**[om\_store,echoes] = EPG\_custom(seq);**

You can view the configuration matrices with om\_store{n}, which corresponds to the n-th event in seq.events and the n-th time in seq.time.

## 2.3 Pre-defined pulse sequences

Pre-defined pulse sequence simulations are title EPGsim\_X.m and their usage can be easily accessed using the help function in MATLAB. Here are some common arguments.

alpha / alphas : flip angle

rlx : relaxation mode, in a 1 x 2 vector [T1,T2] (ms)

TR/esp : repetition time / echo spacing (ms) – related to the spacing between RF pulses

N – number of repeats

All of them share the outputs [om\_store, echoes] where om\_store is a cell array of Ω matrices recorded in time and echoes are nonzero F(0) absolute values and their timings found usually by the function findEchoes. The dimension of echoes is (U x 2) with timing in the 1st column and intensities in the 2nd column.

## 2.4 Displaying EPGs and echoes

display\_epg(om\_store, seq, annot) can be used to visualize simple EPGs (using it to display simulations with too many repeats is not recommended).

To plot echoes, simply plot the output “echoes” with the 2nd column on the y-axis and the 1st column on the x-axis. For an intermediate number of echoes, the stem function can be useful too.

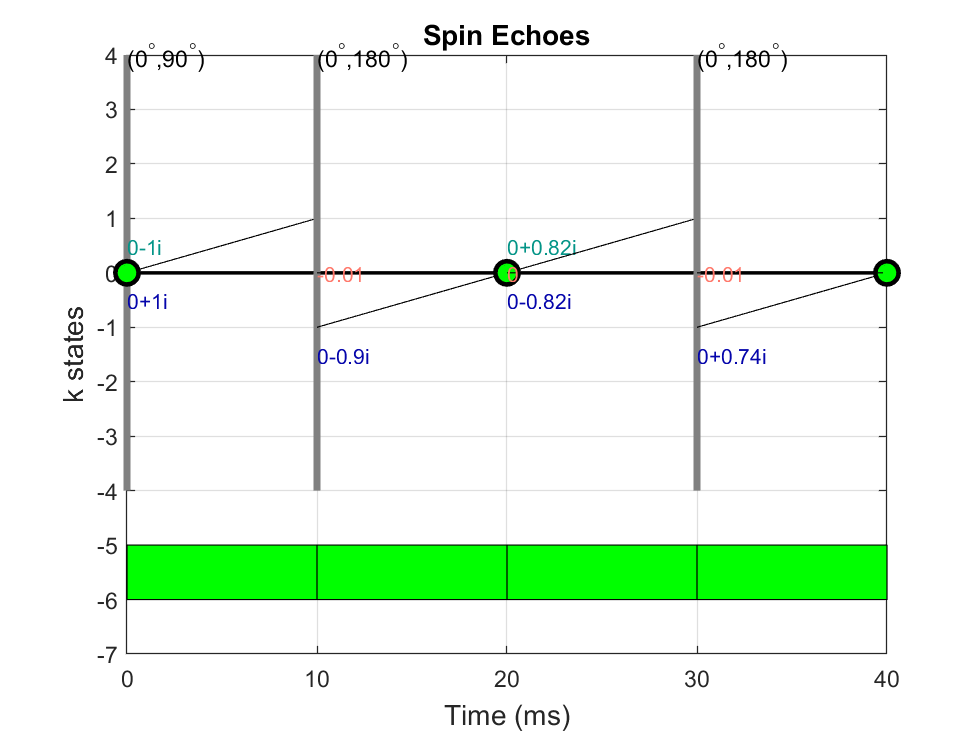


Figure 1: A simple spin echo EPG (example in 2.2)

# 3. References

Weigel, Matthias. "Extended phase graphs: dephasing, RF pulses, and echoes ‐ pure and simple." Journal of Magnetic Resonance Imaging 41.2 (2015): 266-295.

Hennig J, Scheffler K. Hyperechoes. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine. 2001 Jul;46(1):6-12.

----------------------------------------------------------------------------------------------------------------

*Acknowledgement:*

*The author would like to thank Sairam Geethanath, Ph.D., for providing his EPG code as a foundation for the simulation scripts.*

----------------------------------------------------------------------------------------------------------------