



Physics MRI Research Centre

GRADTABLE

Version: 2.10

The Different Types of Gradient Tables and How to Generate Them

Last Updated April 2009 by Zahidur Rahman

Introduction

The various SPRITE-based sequences all use tables of gradient strengths to control the k-space sampling pattern of experiments. These gradient tables are also needed to sort the resulting data, and are used by the simulator program created by James Rioux to numerically reproduce these same experiments. This document describes the important features of the gradient tables currently used in the MRI Centre and also serves as documentation for the program which now is used to generate new tables.

What's upgraded?

In August 2005, The original program was written by James Rioux. Ding Cai added a 2D Sectoral-SPRITE waveform. The original code to generate the waveform was written in Prospa by Alexandr Khrapichev. The version was changed to 1.01

In Sept 2005, Ding Cai added a 3D Sectoral-SPRITE waveform. The original code to generate the waveform was written in Prospa by Alexandr Khrapichev. The version was changed to 1.02.

In March 2006, Biao Wang added a non-standard 48x48x24 3D Sectoral-SPRITE waveform by analyzing the original Prospa code by Alexandr Khrapichev.

In April 2009, the 2D and 3D Sectoral-SPRITE waveform code was found not to work. Zahidur Rahman debugged the program and referred to the original program WAVEGEN2D and WAVEGEN3D to get it to work. The version is changed to 2.10.

Notes on Gradient Tables

Many of the restrictions on the gradients tables in the past have been due to limitations in the software on the acquisition consoles, most notably the Tecmag NTNMR software. For example, no single gradient table can exceed a certain number of characters, so large tables have to be split into multiple segments or have their decimal places truncated. There are IDL programs which can perform either of these tasks.

It is possible to loop through a number of interleaves in a straightforward manner, but only if those interleaves are all the same length and this restriction is unfeasible for any but the simplest two-dimensional sequences. The current workaround is, after calculating the gradient table for a given sequence, to add dummy scans to each interleaf such that their lengths are uniform.

For certain sequences (like the original 39 Cones) this is not possible because of the wide spread in the number of points in various cones. For other sequences, dummy scans can be added, but the amount needed may be large – tens of dummy scans per interleaf. Later in this document, the problem of making the gradient tables more uniform (and therefore minimizing the amount of dummy scans) shall be discussed in more detail.

It is hoped that neither of these problems will manifest on the Resonance Instruments consoles, but as long as the NTNMR software is used, these restrictions will have to be kept in mind when designing new gradient tables.

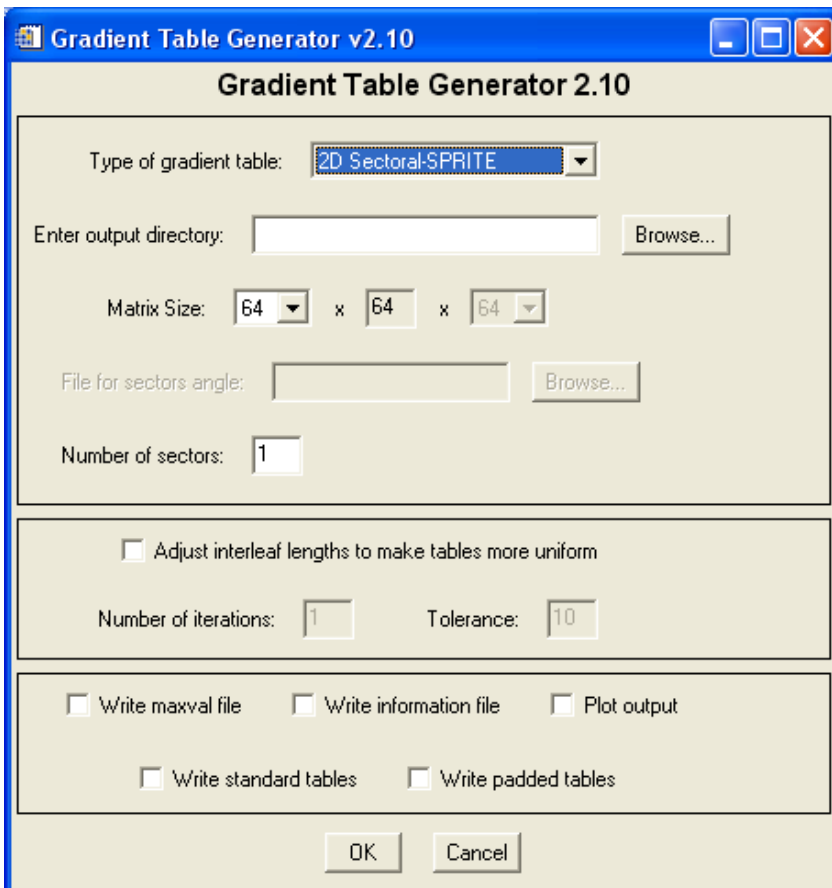
Which Tables to Use?

Most gradient tables come in “non-padded” and “padded” versions, with the exceptions being old Cones sequences, and Spiral-SPRITE with a single interleaf, or 4 interleaves. When using any other kind of sequence...

- In the NTNMR pulse program, use the padded version.
- In the RINMR pulse program, a non-padded version can likely be used.
- In the JR Sim simulator, use the padded version
- When inputting tables to the Universal sorter, use the padded version if the data is from the NTNMR console, and the non-padded version if from the RINMR.

The Gradient Table Generator

This section describes how to use the gradient-table generating IDL program, called gradtable.pro and found in the UNB MRI Programs / Gradient Tables directory on all of the data processing computers. It also explains the algorithm used to calculate the tables, remove duplicate points, and make padded tables more uniform if desired.



All of the information needed to create the initial gradient waveform is located in the top half of the dialog box shown above. After choosing the type of gradient tables, which can be (2D Spirals, 2D Radial, 2D Sectoral, Standard Conical, Interleaved Conical, 3D Radial, 3D Spherical or 3D Sectoral)-SPRITE the options available below will be modified accordingly.

Some sequences use, the “number of interleaves” or “number of cones” text fields to enter the appropriate values. With interleaved cones, these are replaced by another text field for a file containing interleaf numbers. This file should contain one number on each line to denote the number of interleaves desired for each cone in the sequence. An example, 39cones_numleaves.txt, has been provided for the case of 39 interleaved cones. These file are stored in UNB MRI Programs\gradtable\Sector Angles folder for use with gradtable.

The “keep duplicates” field should be set to a value between 0 and the matrix size. Any points which are on different interleaves but have the same T_1 weighting will be kept if they fall within this distance from the center of k-space.

When complete, the resulting files will be written to the folder specified in the “Output directory” text field. The filenames are generated automatically based on the type of gradient table, the matrix size selected, and the number of points in the final table.

The following options are available when deciding what kind of output to generate:

- Write standard tables (writes the basic X, Y, Z gradient tables)
- Write padded tables (writes tables such that every interleaf is the same length)
- Write maxval file (writes an additional file with numbers of points for each leaf)
- Write information file (writes an additional file with even more information)
- Plot output (generates a 2D or 3D rendering of the gradient waveforms)

When testing out new gradient tables, it is recommended that only the plot option be chosen until the waveform looks correct; then the files can be written. Also note that the plotting may take some time for large 3D waveforms with multiple interleaves and if it too large it might crash while processing the request.

Generating the Tables

All types of interleaved gradient tables are generated in the same way – start with a single interleaf given by the appropriate equations, and rotate that interleaf multiple times until enough interleaves have been generated. In the case of cones, different equations must be used for each cone (with the angle of inclination varying from +90 to -90 degrees) but any interleaves within a cone are generated by rotation as described above.

All the waveforms are initially generated with floating-point numbers to create smooth trajectories, but they are then gridded to integer values. (This allows the sampled data to be reconstructed with an FFT.) This creates many duplicate points, both within the same

interleaf and across interleaves. Under no circumstances do we want the same location to be sampled twice on the same interleaf, so all duplicate points on the same interleaf are removed before proceeding further.

Duplicate points between interleaves are allowed, but only if the T_1 weighting is the same at both points – that is, both points are at the same location along the waveform. If the same point is visited at two different times on two different interleaves, it is retained only on that interleaf where it occurs earliest. This ensures that the T_1 weighting is kept as close to its original (possibly prepared) value as possible.

For 3D sequences, usually only the upper half of the tables are generated (i.e. for $z > 0$). The tables are then flipped about $z=0$ to fill the lower half of k-space. This means we only have to process duplicate points for half the interleaves, reducing computation time.

Making Tables More Uniform

Until now we have overlooked the options in the middle portion of the gradtable.pro dialog box, which try to make the interleaf lengths more uniform. These shall now be described in more detail, along with the length-uniforming algorithm.

The problem we are trying to rectify is that, though all of the interleaves are initially the same length, after the removal of duplicate points some interleaves will invariably be longer than others – much longer, in some cases. Since every interleaf must be the same length in gradient tables used on the Apollo console, a large amount of dummy scans have to be added to the shorter interleaves to make all the interleaf lengths identical. This can increase file sizes by a considerable amount (over 10%).

To circumvent this situation, or at least minimize its impact, an algorithm has been developed that attempts to make the interleaf lengths more uniform before dummy scans are added. This is done by removing points from the end of long interleaves and adding them to the end of shorter ones.

After the removal of duplicate points, the average interleaf length is calculated, and any interleaves that exceed or fall short of this average length by more than a certain amount (the “tolerance” value entered in the dialog box) are marked for processing. The program will then try to find two interleaves which are on opposite sides of the average - say, one that has 40 points, another has 58, and the average is 50 – and whose endpoints are close to one another. Points are taken from the longer interleaf and added to the end of the shorter interleaf so that their lengths are equal (and hopefully now close to the average).

Ideally, this continues until all interleaves are closer to the average than the tolerance value provided. The lower the tolerance value, the more uniform the gradient tables should become. In some cases a second pass through the gradient tables may be needed; if this is desired, change the “number of iterations” field to 2 instead of 1. (More than 2

iterations should never be needed.) The program will re-calculate the average with the modified tables and run the algorithm again.

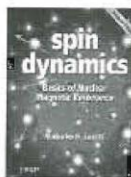
It is still possible for there to be one or two longer interleaves, even after this procedure. As a final attempt to fix this, any interleaves which are 15% longer than the average, or more, will be truncated to the average length before any padded tables are written. This is only done if the “write padded tables” option in the dialog box has been checked off.

Miscellaneous

Sometimes a gradient table is too long to fit into the appropriate table in the NTNMR pulse program; if this happens the table can be split into several smaller tables using `split_tables.pro`, or the precision of the numbers can be reduced using `reduce_table.pro` to make the overall size smaller. Both these programs are in the UNB MRI Programs\gradtable\Helping program folder.

For 3D Sectoral SPRITE waveform described in Alexandre A. Khrapitchev’s JMR paper “Sectoral sampling in centric-scan SPRITE magnetic resonance imaging” we employ sector angle files to help create waveforms. These files are originally from the website of Malcolm H. Levitt, <http://www.mhl.soton.ac.uk/public/Main/index.html>. The important and relevant section of the website is attached to this manual. There are many different types of sector angle file in his website and they possess 3 column format but we use only first 2 columns in gradtable program. The ones that are currently used by MRI Centre are LEB50, LEB146, LEB302, LEB590, SHREWD_REP150 and 700 and ZCW55, ZCW144 and ZCW616 where the letter refer the name of the angle set and the number refer the number of orientations. These files were already extracted from the website and processed by taking out 3rd column and saved as 2 columned file and stored in a sub folder of gradtable program folder (UNB MRI Programs\gradtable\Sector Angles). Some of the gradient tables produced are too large and the processor cannot handle the operation and just crash on executing time. As for example we can create a gradient table of a matrix size 64*64*64 with 700 sectors but if we try to create 128*128*128 with 50 sectors it just crash and not respond.

[Levitt
group
home](#)



[Spin
Dynamics
2nd
edition](#)



[Research
programs](#)

[software](#)

[NMR
topics
\(Euler
angles,
composite
pulses
etc.\)](#)

[publication:](#)

[teaching](#)

[links](#)

[local
group
information](#)

[local
NMR
user](#)

Orientations for Powder Averaging



Sets of orientational angle files for powder averaging of solid-state NMR spectra. The sets include some of the Gaussian Quadrature Schemes described in ref.96. There is also a picture gallery of some of the sets.

There are sets available for 3-angle averaging using the ZCW method, and for 2-angle averaging over a [hemisphere](#), an [octant](#), and over the [full sphere](#). We include the [LEBoct](#) and [LEBhemi](#) sets recommended in ref.96, as well as some [ZCWoct](#) and [REPULSION](#) sets for comparison.

[Laikov 2-angle sets](#)

[Karney 3-angle sets](#)

NEW [POLYTOPE 3-angle sets](#)

Additional orientational angle sets, including ROSE sets, are available on [Mattias Edén's site](#).

The following sets are available:

3-angle sets

These are suitable for powder averaging of functions without any orientational symmetry. They are given in the form {alpha,beta,gamma,weight}. The angles {alpha,beta,gamma} are in **radians**. The following sets are available:

- o **zcw3**: 3-angle orientational sets, generated by the ZCW algorithm (also known as the Conroy or Cheng method -- see [references](#)). We recommend these for orientational averaging of functions which do not possess orientational symmetry. (Usually much more efficient than using a 2-angle set and stepping through the third angle).

number of orientations= [50](#), [100](#), [144](#), [200](#), [300](#), [538](#), [1154](#), [3722](#), [6044](#)

- o **Karney sets**: 3-angle orientation sets, generated by [Charles Karney](#) (Sarnoff corporation) using quaternion algebra. For the derivation of these sets, see Charles Karney's website and [this article](#). These sets have not yet been tested for solid-state NMR simulations, but it is likely that they are superior to the ZCW sets as they are based on a more rigorous treatment of 3D rotational space. Please give me [feedback](#) on any experiences you have with these sets. Download a zipped archive of all Karney sets [here](#).

Note: These sets have been converted from quaternion form to

Note: These sets have been converted from quaternion form to Euler angle form and published on this website with the permission of Charles Karney. If you use these sets for published work, please reference Charles Karney.

- **Lebedev 3-angle sets:** see [Mattias Edén's site](#).
- **POLYTOPE sets:** generated from 4-dimensional regular polytopes. See [this article](#).
 - [POLYTOPE60](#)

2-angle sets

These are suitable for averaging of functions possessing orientational symmetry. They are given in the form {alpha,beta,weight}. The angles {alpha,beta} are in **radians**. The circumstances under which these sets may be used in NMR are described in [ref.96](#). The sets are classified according to the Ci or D2h symmetry of the averaged function.

More 2-angle sets are available on [Mattias Edén's site](#).

Ci sets (averaging over hemisphere)

- **LEBhemi:** Lebedev sets for 2-angle averaging over a hemisphere. See [references](#).
number of orientations= [25](#), [43](#), [55](#), [73](#), [97](#), [151](#), [295](#), [385](#)
- **STEPhemi:** stepping through 2-angles over a hemisphere, using sin(beta) weighting.
number of orientations= [36](#), [64](#), [100](#), [144](#), [256](#), [324](#), [576](#), [900](#)
- **SHREWD_STEPhemi:** stepping through 2-angles over a hemisphere, using weighting optimized for Gaussian Spherical Quadrature. The large sets may often be used to obtain fully converged "reference" spectra.
number of orientations= [66](#), [120](#), [190](#), [276](#), [378](#), [496](#), [630](#), [780](#), [946](#), [1128](#), [1326](#), [3321](#), [5151](#)

D2h sets (averaging over octant)

- **LEBoct:** Lebedev sets for 2-angle averaging over an octant. See [references](#). We recommend these sets for calculation of MAS sb patterns of isolated spins.
number of orientations= [10](#), [16](#), [19](#), [22](#), [31](#), [37](#), [46](#), [85](#), [109](#)
- **STEPoct:** stepping through 2-angles over an octant, using sin(beta) weighting.
number of orientations= [25](#), [36](#), [49](#), [81](#), [100](#), [144](#), [225](#), [361](#), [729](#), [900](#)

- **SHREWD_STEPoct**: stepping through 2-angles over an octant, using weighting optimized for Gaussian Spherical Quadrature. The large sets may often be used to obtain fully converged "reference" spectra. number of orientations= [18](#), [50](#), [66](#), [72](#), [98](#), [128](#), [190](#), [231](#), [288](#), [392](#), [496](#), [648](#), [861](#), [1058](#), [1326](#)
- **ZCWoct**: 2-angle orientational sets, generated by the ZCW algorithm over an octant (also known as the Conroy or Cheng method -- see [references](#)). number of orientations= [21](#), [34](#), [55](#), [89](#), [144](#), [233](#), [377](#), [616](#), [987](#)
- **SHREWD_ZCWoct**: Improved 2-angle orientational sets, choosing the weights according to Gaussian Spherical Quadrature. number of orientations= [21](#), [34](#), [55](#), [89](#), [144](#), [233](#), [377](#)

full sphere sets

These are included for completeness. There are no known NMR problems which require only 2-angle averaging which do not also fall into the Ci or D2h symmetry groups.

LEB: Lebedev sets for 2-angle averaging over a full sphere. See [references](#).

number of orientations= [50](#), [86](#), [110](#), [146](#), [194](#), [302](#), [590](#), [770](#)

NEW [C-program](#) by Dmitri Laikov (Moscow State University, Moscow, Russia), slightly modified by Marina Carravetta, for generating two-angle Lebedev-Laikov orientation sets up to maximum rank =131 (5810 orientations).

NEW [Extended compilation of Lebedev orientations and weights](#), compiled by Marina Carravetta.

- **REP**: REPULSION sets, supplied by M. Bak and N. C. Nielsen, see *J. Magn. Reson.* **125**, 132 (1997)). *Please acknowledge these authors if you use these orientations* number of orientations= [100](#), [150](#), [168](#), [232](#), [376](#), [700](#)
- **SHREWD_REP**: Improved REPULSION sets, with weights calculated by the Gaussian Spherical Quadrature. number of orientations= [100](#), [150](#), [168](#), [232](#), [376](#), [700](#)
- **ZCW** 2-angle orientational sets, generated by the ZCW algorithm over the full sphere (also known as the Conroy or Cheng method -- see [references](#)). number of orientations= [21](#), [34](#), [55](#), [89](#), [144](#), [233](#), [377](#), [616](#), [987](#)
- **SHREWD_ZCW**: Improved 2-angle ZCW sets, choosing the weights according to Gaussian Spherical Quadrature. number of orientations= [21](#), [34](#), [55](#), [89](#), [144](#), [233](#), [377](#), [616](#)