## Contents

<table>
<thead>
<tr>
<th>Chapter 1</th>
<th>Introduction</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>About this manual</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>About Shapes</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>Conventions</td>
<td>6</td>
</tr>
<tr>
<td>Chapter 2</td>
<td>Interactive Shapetool</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Opening a Shape/Gradient</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Using Shape Tool Display Options</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Saving a Shape/Gradient as a 1D dataset</td>
<td>9</td>
</tr>
<tr>
<td>2.4</td>
<td>Superimposing Multiple Shapes/Gradients</td>
<td>9</td>
</tr>
<tr>
<td>2.5</td>
<td>Saving a Shape/Gradient</td>
<td>10</td>
</tr>
<tr>
<td>2.6</td>
<td>Generating Shapes/Gradients</td>
<td>10</td>
</tr>
<tr>
<td>2.7</td>
<td>Analysing Shapes</td>
<td>12</td>
</tr>
<tr>
<td>2.8</td>
<td>Manipulating Shapes</td>
<td>18</td>
</tr>
<tr>
<td>2.9</td>
<td>Setting Shape Tool Options</td>
<td>24</td>
</tr>
<tr>
<td>2.10</td>
<td>Examples of using the Shape Tool</td>
<td>26</td>
</tr>
<tr>
<td>Chapter 3</td>
<td>Generating Shapes with st</td>
<td>33</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Basic Shapes</td>
<td>34</td>
</tr>
<tr>
<td>3.3</td>
<td>Classical Shapes</td>
<td>37</td>
</tr>
<tr>
<td>3.4</td>
<td>Adiabatic Shapes</td>
<td>42</td>
</tr>
<tr>
<td>3.5</td>
<td>Special Shapes for Solids Applications</td>
<td>46</td>
</tr>
<tr>
<td>3.6</td>
<td>Special Shapes for Imaging Applications</td>
<td>47</td>
</tr>
<tr>
<td>3.7</td>
<td>Special Shapes for Decoupling</td>
<td>48</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>Manipulating Shapes with st</td>
<td>49</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>49</td>
</tr>
<tr>
<td>4.2</td>
<td>Command offs</td>
<td>50</td>
</tr>
<tr>
<td>4.3</td>
<td>Command sinm2</td>
<td>52</td>
</tr>
<tr>
<td>4.4</td>
<td>Command cosm2</td>
<td>52</td>
</tr>
<tr>
<td>4.5</td>
<td>Modulation according to Frequency Sweep</td>
<td>52</td>
</tr>
<tr>
<td>4.6</td>
<td>Command power</td>
<td>53</td>
</tr>
<tr>
<td>4.7</td>
<td>Command scale</td>
<td>53</td>
</tr>
<tr>
<td>4.8</td>
<td>Command addphase</td>
<td>54</td>
</tr>
<tr>
<td>4.9</td>
<td>Command trev</td>
<td>54</td>
</tr>
<tr>
<td>4.10</td>
<td>Command expand</td>
<td>55</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Analyzing Shapes with st</td>
<td>57</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>57</td>
</tr>
</tbody>
</table>
Chapter 5

5.2 Command bandw2 .................................................. 58
5.3 Command bandw2i .................................................. 58
5.4 Command bandw2ry .................................................. 59
5.5 Special Bandwidth Calculations ................................. 59
5.6 Command calcb1mo ................................................. 59
5.7 Command calcb1adia .............................................. 60
5.8 Command bsiegert3 ............................................... 60
5.9 Command calcpav .................................................. 61
5.10 Command integr3 .................................................. 61
5.11 Command integradia ............................................. 62
5.12 Routines which have been replaced in stdisp ............... 62

Chapter 6 Miscellaneous st functions ................................. 65

6.1 Introduction ....................................................... 65
6.2 Commands convert, convertgr ................................. 65
6.3 Command add ..................................................... 66

Chapter 7 Shapes in AU programs ................................ 67

7.1 Using Shape Tool commands in AU Programs .............. 67

Chapter 8 Shape File Format ......................................... 69

8.1 Format of a Gaussian Shape File ............................... 69
Chapter 1
Introduction

1.1 About this manual

This manual describes the Topspin Shapetool, the program which allows you to create, analyze and manipulate RF- and Gradient Shapes. Shape-tool can be used in two different ways:

• The command stdisp opens a graphical interface that allows to view shapes, and to interactively create and manipulate shapes using various parameters.

• The command st <parameters> allows to generate or manipulate shapes from the TOPSPIN command line. The main purpose of this syntax is that the command can easily be included in AU programs to produce shapes automatically.

Chapter 2 of this manual describes the graphical interface, whereas the remaining chapters describe the usage of the st command.

1.2 About Shapes

Shapetool allows you to create, analyze or manipulate two types of shapes, RF shapes and gradients. An RF shape consists of phase and
amplitude values, whereas a gradient only consists of amplitude values. Once created or changed, a shape is stored in a text file in JCAMP-DX format suitable to be displayed with stdisp or executed by TOPSPIN acquisition commands.

For RF shapes all files are read from or stored in the directory:

<topspinhome>/exp/stan/nmr/lists/wave

For gradients all files are read from or stored in the directory:

<topspinhome>/exp/stan/nmr/lists/gp

1.3 Conventions

**Font conventions**

*stdisp* - commands to be entered on the command line are underlined

*Analysis* - commands to be clicked are in times bold italic

*Alignment* - Dialog window fields are in times bold

*intrng* - filenames are in courier

*name* - any name which is not a filename is in times italic
To start the interactive Shapetool interface:

+ Click *Spectrometer ’ Shape Tool*

or enter `stdisp` on the command line.

The Shape Tool window will appear (see Fig. 2.1.). This consists of a tool-
bar, a command line and a split pane with a data section at the right and a
parameter section at the left.
By default, a 1000 point Gauss shape is displayed with Truncation level 1.0.

The TOPSPIN menu is changed showing the additional *Shapes* and *Manipulate* menus and the adjusted *File*, *Analysis* and *Options* menus.

Note that all functions of the interactive Shape Tool can also be performed non-interactively with the TOPSPIN command `st`. This command must entered with the appropriate arguments on the command line while the associated dataset is displayed and selected. For a description of the `st` command see:

```
+ Help ' Application manuals ' Shape Tool
```

### 2.1 Opening a Shape/Gradient

To open an existing shape/gradient, click the button:

![Open a shape or gradient.](image)
select *Shape* or *Gradient*, select a file from the appearing file list and click *OK*. The selected shape/gradient will appear in the data section.

### 2.2 Using Shape Tool Display Options

The Shape Tool toolbar offers the following display options:

- Display amplitude and phase.
- Display amplitude only.
- Display phase only.
- Toggle cursor information on/off. Cursor information consists of the point number and amplitude/phase value.
- Wrap/unwrap phase (actual phase/phase modulo 360°).
- Toggle between polar and cartesian coordinates.

### 2.3 Saving a Shape/Gradient as a 1D dataset

To save the current shape/gradient stored under a procno of the associated dataset, click:

- **1d** Save shape/gradient as a 1D dataset.

### 2.4 Superimposing Multiple Shapes/Gradients

Several shapes/gradient can be displayed superimposed using the following buttons:

- Add the currently displayed shape to multiple display.
- Switch to multiple display mode. This shows all 'added' shapes/gradients superimposed.
2.5 Saving a Shape/Gradient

To save a shape/gradient, click the button:

![Save a Shape or Gradient or a fraction of it.]

and select *Save Shape*, *Save Gradient* or *Save Fraction*. A dialog window appears where you can enter a (File)*Name, Title, Flip Angle* and the *Type of Rotation*.

![Figure 2.2](image)

If you save a fraction of a shape or gradient, the above dialog is preceded by the following dialog:

![Enter the Start and End point and click OK.](image)

2.6 Generating Shapes/Gradients

The Shape Tool allows you to generate RF and gradients shapes. Various
Basic, Classical or Adiabatic shapes can be created from the Shapes menu. Additionally Solids, Imaging and Decoupling shapes are available.

How to Generate a Gauss shape
To generate a Gauss shape:

+ Click Shapes ’ Classical Shapes ’ Gauss

A Gaussian shaped curve will appear in the data section. The parameter section shows the two parameters that can be set for a Gauss:

Size of Shape: the number of shape data points

Truncation level: the minimum amplitude at the edge of the shape

If you change these parameters, the displayed shape will automatically be updated. To save the shape, click , enter a Name, e.g. mygauss, Title, Flip Angle and the Type of Rotation (see chapter 2.5) and click OK.

How to Generate a ShapFour shape
To generate an RF shape that is defined by Fourier coefficients:

+ Click Shapes ’ Classical Shapes ’ ShapFour

The following dialog box will appear:

![Parameters for ShapFour](image)

Figure 2.3

Here you can enter the Size of Shape and the Number of coefficients. Clicking OK or Apply will open a further dialog box (see Fig. 2.4), where you can enter the required coefficients. A ShapFour shape is defined by two coefficients arrays a[0,1,...] and b[1,...]. Note that the element a[0] is list-
ed separately.

![ShapFour Coefficients](image)

**Figure 2.4**

Here, you can enter the desired coefficients and then click:

*Save* to store the coefficients for later usage. You will be prompted for a filename which will be stored as:

```
<user-home>/topspin-<hostname>/shapetool
```

*Apply* to display the shape while keeping the dialog box open for possible further changes.

*OK* to display the shape and close the dialog box.

With the coefficients above you have created a *Rsnob* shape.

To save it:

Click ![click](image), enter a *Name*, e.g. *myrsnob*, *Title*, *Flip Angle* and the *Type of Rotation* (see chapter 2.5) and click *OK*.

### 2.7 Analysing Shapes

The Shape Tool interface offers several functions to analyse shapes. Most of these functions are only meaningful for RF shapes.

To access these functions:
+ Click **Analysis**

in the **TOPSPIN** menu bar. This will open the pull-down menu shown in Fig. 2.5.

<table>
<thead>
<tr>
<th>Calculate Bandwidth for Excitation [analyze bandw2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate Bandwidth for Inversion [analyze bandw2i]</td>
</tr>
<tr>
<td>Calculate Bandwidth for Refocusing -My [analyze bandw2ry]</td>
</tr>
<tr>
<td>Special Bandwidth Calculations</td>
</tr>
<tr>
<td>Calculate gammaB1 max [analyze calcb1mo]</td>
</tr>
<tr>
<td>Calculate gammaB1 max for Adiabatic Shapes [analyze calcb1adia]</td>
</tr>
<tr>
<td>Calculate Bloch-Siegert Shift [analyze bsiegert3]</td>
</tr>
<tr>
<td>Calculate average Power Level [analyze calcpav]</td>
</tr>
<tr>
<td>Integrate Shape [analyze integr3]</td>
</tr>
<tr>
<td>Integrate Adiabatic Shape [analyze integradia]</td>
</tr>
<tr>
<td>Integrate and compare to Reference [analyze integrandcomp]</td>
</tr>
<tr>
<td>Simulation [analyze simulate]</td>
</tr>
</tbody>
</table>

**Figure 2.5**

From here, you can start the analysis functions discussed below.

**How to Calculate the Bandwidth for Excitation**

To calculate the excitation bandwidth factor \( \Delta \omega \Delta T \):

+ click **Analysis’ Calculate Bandwidth for Excitation** [analyze bandw2]

For the excitation bandwidth, the \( (M_x^2 + M_y^2) \) magnetization is used.

Fig. 2.6 shows the resulting parameter section for a Gauss shape.
As you can see, for a 90° Gaussian shape the bandwidth factor is 2.122.

The bandwidth factor is the product of the width of excitation (\(\Delta\omega\)) and the pulse length (\(\Delta T\)).

This means a pulse length of 21220 \(\mu\text{sec}\) gives a bandwidth (\(\Delta\omega\)) of 100.0 Hz. The bandwidth is the excitation width at the 3 dB point, i.e. the point where the magnetization has dropped to 70.8 %.

When you change one of the parameters in the parameter section, the others are automatically adjusted. Clicking the Update Parameters button will store the length of the shaped pulse (\(\Delta T\)) to the associated dataset parameter (default P11).

**How to Calculate the Bandwidth for Inversion**

To calculate the inversion band width factor \(\Delta\omega \Delta T\):

+ Click *Analysis ’ Calculate Bandwidth for Inversion* [analyze bandw2i]

For the inversion band width, the Mz magnetization is used.

**How to Calculate Bandwidth for Refocusing**

To calculate the refocusing band width factor \(\Delta\omega \Delta T\):
Interactive Shapetool

+ Click Analysis’ Calculate Bandwidth for Refocusing [analyze bandw2ry]

For the refocusing band width, the -My magnetization is used.

How to Calculate Special Bandwidths

Further routines for the evaluation of various magnetization components are available under the menu item Special Bandwidth Calculations.

How to Calculate the Maximum RF Field Strength for Classical Pulses

To calculate the RF field strength for classical pulses:

+ Click Analysis’ Calculate gammaB1max [analyze calcb1mo]

Fig. 2.7 shows the resulting parameter section for a Gauss shape.

![Gauss parameters](image)

**Figure 2.7**

The result consists the field strength (γB1max/2π) and the corresponding 90° square pulse length. To obtain this result, the integral of the shaped pulse is compared to that of a square pulse of same length, with the latter being normalized to 1.

The maximum power is:

γB1(max)/2π = γB1/2π (square pulse of same length) / integral ratio

where
\( \gamma B_1/2\pi \) (square pulse of same length) = (1 / pulse length * (360°/ flip angle))

**How to Calculate the Maximum RF Field Strength for Adiabatic Pulses**

To calculate the RF field strength for adiabatic pulses:

+ Click *Analysis* ’ *Calculate gammaB1max for Ad. shapes* [analyze calcb1adia]

![HypSec](image)

**Figure 2.8**

Fig. 2.8 shows the resulting parameter section for a *HypSec* shape.

The result \( \gamma B_1(\text{max})/2\pi / \sqrt{(Q)} \) is calculated from the on-resonance sweep rate.

Q(t) is the quality or adiabaticity factor during the shape. After entering the appropriate value for Q with respect to the middle of the shape, the value for \( \gamma B_1(\text{max})/2\pi \) is obtained. As a rule of thumb Q is set to 5 for inversion pulses. However, for decoupling pulses, Q should be set between 2 and 3 to allow a lower decoupling power.

**How to Calculate the Bloch-Siegert Shift**

To calculate the phase difference due to the Bloch-Siegert shift:

+ Click *Analysis* ’ *Calculate Block-Siegert Shift* [analyze bsiegert3]
Input parameters are the length of the shaped pulse in $\mu s$, $\gamma B1(\text{max})$ in Hz and the offset of a theoretical signal relative to the frequency of the RF-pulse in Hz. The result consists of the phase difference due to Bloch-Siegert shift in Degree and the corresponding frequency shift in Hz.

**How to Calculate the Average Power Level**

To calculate the average power:

+ Click *Analysis’ Calculate average Power Level* [analyze calcpav]

The result is the percentage of the average power of a square pulse of the same length.

**How to Integrate Classical Shapes**

To calculate the power level required for a classical shaped pulse:

+ Click *Analysis’ Integrate Shape* [analyze integr3]

![Gauss](image)

**Figure 2.9**

Fig. 2.9 shows the parameter section where you can enter the length of the soft pulse, the flip angle and the length of the 90° hard pulse. Soft pulse and hard pulse length are initialized with the values of the corresponding parameters of the associated dataset. If these values are zero
or if no dataset is associated, default values are taken, a 10000 µs soft pulse and 50 µs hard pulse.

The result for a Gaussian shape with 1000 points and 1% truncation is shown. The *Integral ratio* and the *Corresponding difference* in dB are calculated from shape amplitude and phase, assuming equal pulse length and flip angle for soft and hard pulse. The *Change in power level*, however, is calculated from the specified pulse lengths and rotation.

Clicking the *Update Parameters* button saves the length of the shaped pulse and the change in required power level to the corresponding parameters of the associated dataset. Note that the latter is the sum of the shown *Change of power level* and the current hard pulse power level (e.g. PL1).

**How to Integrate Adiabatic Shapes**

To calculate the power level required for an adiabatic shaped pulse:

+ Click *Analysis ’ Integrate Adiabatic Shape* [analyze integradia]

The change of power level is calculated from the length of the corresponding 90° pulse and the length of the hard pulse.

Clicking the *Update Parameters* button saves the length of the shaped pulse and the required power level to the corresponding parameters of the associated dataset.

**How to Start Simulation**

To evaluate the shape behaviour, you can start the TOPSPIN NMRSIM routine:

+ Click *Analysis ’ Simulation* [analyze simulate]

This allows you for example, to view the excitation profile, or, for adiabatic pulses, check the adiabaticity.

**2.8 Manipulating Shapes**

Shapes can be manipulated in various ways. Most of these functions are only meaningful for RF shapes.
Frequency Encoding

To perform a phase (and amplitude) modulation encoding one or more offset frequencies:

+ Click *Manipulate ’ Phase modulation acc. to Offset Freq.* [manipul offs]

Figure 2.10

A dialog box will appear (see Fig. 2.10) where you can set the required parameters.

The *Alignment* radio buttons allow you to define the position of phase 0; the beginning, middle or end of the shape.
The **Reference Frequency** radio buttons allow you to choose whether the reference frequency:

- is not used
- is set to O1 of the associated dataset
- is set to the first entry of the frequency list. The **Option** '
  Frequencies taken from Frequency' List will automatically be checked.

The **Options** check boxes allow you to set:

- *Frequencies taken from a Frequency list*. The **Parameter** Name of Frequency List will automatically appear. Frequency lists can be set up from the TOPSPIN interface with the command edlist or, interactively, by clicking the button.
- *With additional Phase Setting*
- *With additional Scaling (0-100%):* the relative amplitude for each excitation region. The overall amplitude will be divided by the number of frequencies. For example, for 3 frequencies with equal scaling factors, each frequency will contribute 33%.

The **Parameters** fields:

- The Length of Pulse
- The **Number of Frequencies** or, when the **Option** Frequencies taken from a Frequency list is checked, Name of the Frequency List.

Clicking **Apply** or **OK** will open a new dialog box where the selected parameters are shown.

If no reference frequency is used, the values entered as frequencies have to be difference frequencies.

**How to Calculate a Shape from an Excitation Region**

A shape can be modulated according to specific regions in a spectrum. To do that, take the following steps:

1. Switch to the associated dataset.
2. Determine the integral regions interactively.
3. Switch back to the Shape Tool interface.
4. Click **Manipulate ’ Calc. Shape from excitation Region**
If the intrng file does not exist, you will be prompted for the number of regions and the carrier frequency $O_1$. In that case, the fields **Left Limit** and **Right Limit** are empty. Note that overlapping regions are not allowed.

In Fig. 2.11 the default values for Flip Angle, Initial Phase, Alignment and Type of 180 Degree Pulse are shown. Depending on the application these settings have to be modified accordingly.

You can use the same shape for each region of all different shapes. In the latter case, the check box *Use same Shape for all Regions* must be unchecked.

**How to Add Shapes**

To add multiple shapes:
+ Click **Manipulate ’ Add Shapes** [manipul addshapes]

A dialog box will appear where you can select up to 10 shapes from the available shape files.

A minimum of two shapes must be selected. Note that the selected shapes replace the current shape.

The file size of each shape is shown in the *Size* column. Note that if the sizes differ, the desired alignment must be selected.

For each added shape, you can specify a scaling factor between 0 and 100% in the *Scaling* column.

The *Scale resulting shape* radio button allows you to rescale the resulting shape:

- to 100%
- to the average of the individual scaling factors

The alignment of the added shapes may be with respect to the *Beginning of Shape*, the *Center of Shape* or the *End of Shape*. If *Arbitrary Alignment* is selected, the start point must be entered for each shape in the *Start* column. In other cases the *Start* entry is deactivated.

### How to Perform Single Sine Modulation

To perform an amplitude modulation such that the pulse excites at two symmetric sidebands (+/- offset) with opposite phase:

+ Click **Manipulate ’ Single Sine Modulation** [manipul sinm2]

and set *Offset frequency* to be relative to the carrier frequency.

### How to Perform Single Cosine Modulation

To perform an amplitude modulation such that the pulse excites at two symmetric sidebands (+/- offset) with the same phase:

+ Click **Manipulate ’ Single Cosine Modulation** [manipul cosm2]

and set *Offset frequency* to be relative to the carrier frequency.

### How to Create a Shape with a Linear Sweep

To create a shape with a phase modulation according to a linear frequency sweep:
+ Click *Manipulate’ Linear Sweep* [manipul sweep]

and set the pulse length and sweep width. Then specify the Q factor to obtain the value for $\gamma B1(\text{max})/2\pi$. Q is the quality or adiabaticity factor which, as a rule of thumb Q is set to 5 for inversion pulses and between 2 and 3 for decoupling pulses.

**How to Create a Shape with Const. Adiabaticity Sweep**

To create a shape with a phase modulation according to a constant adiabaticity sweep:

+ *Manipulate’ Const. Adiabaticity Sweep’* [manipul caSweep]

and set the pulse length and sweep width. Then specify the Q factor to obtain the value for $\gamma B1(\text{max})/2\pi$. Q is the quality or adiabaticity factor which, as a rule of thumb Q is set to 5 for inversion pulses and between 2 and 3 for decoupling pulses.

**How to Calculate a Shape with Amplitude to the Power x**

To create a shape with an amplitude to the power of a specified exponent:

+ Click *Manipulate’ Power of Amplitude* [manipul power]

and enter the desired exponent.

**How to Scale the Amplitude**

To scale the amplitude of a shape to a specified percentage:

+ Click *Manipulate’ Scale Amplitude* [manipul scale]

and enter the desired scaling factor (percentage).

**How to Add Constant Phase**

To add a constant phase to the shape:

+ Click *Manipulate’ Add constant Phase* [manipul addphase]

and enter the phase to be added.

**How to Perform Time Reversal**

To time reverse a shape:

+ Click *Manipulate’ Time Reversal* [manipul trev]
How to Expand Shape

To expand a shape according to a specified phase list (supercycle):

+ Click *Manipulate ’ Expand Shape* [manipul expand]

and selected a phase cycle. The following standard phase cycles are offered:

- mlev4, mlev16, p5(150 deg), p5(330 deg), p5m4, p5m16, p9, p9m16, p5p9

For example, *mlev4* has the following supercycle: 0 0 180 180

In addition, the entry *YourOwn* appears, which allows you to define your own phase cycles. These can be saved for later usage. The specified file must have the extension .expand. It is stored in the directory:

<user-home>/topspin-<hostname>/shapetool

2.9 Setting Shape Tool Options

When the Shape Tool window is selected, the upper three entries in the *TOPSPIN Options* menu concern the Shape Tool.

How to Change the Relation between Shape Tool and TOPSPIN parameters

To define the relation between Shape Tool parameters and TOPSPIN acquisition parameters:

+ Click *Options ’ Define Parameter Table*
A dialog box will appear with a list box for each parameter (see Fig. 2.12).

![Define Parameter Table]

**Figure 2.12**

Make any changes you want and click *OK*. The acquisition parameters of the associated dataset can be set/viewed from the TOPSPIN interface by clicking *AcquPars* or entering *eda*.

**How to Change the Shape Storage Directory**

To change the shape storage directory:

+ Click *Options ’ Set path to Shape Directory*
In the appearing dialog box, you can select the default path or select and specify a different path. Note that a shape base directory must be \texttt{shape} and a gradient base directory must be \texttt{gp}. Opening and saving shapes/gradients will access the specified directory.

**How to select the Associated Dataset**

To set the dataset associated to the Shape Tool interface:

- Click \textit{Options} \textit{Select associated dataset}

A list of open datasets will appear. Select the desired dataset and click \textit{OK}.

### 2.10 Examples of using the Shape Tool

**Example 1: Create a Shaped Pulse for Multiple Solvent Suppr. using WET**

To create a shaped pulse:

1. Acquire a reference 1D spectrum.
2. Click \作坊 \作坊 to switch to frequency list mode.
   - Select the frequency list type and enter the list name \textit{freqlist}. If you get a message that this list already exists, click \textit{Overwrite} to overwrite the existing list or click \textit{Cancel} and specify a new name.
• Move the vertical cursor line to the desired frequencies.
• Left-click the desired frequencies.
• Click to save the frequency list and return.

3. To view the frequency list:
   + Enter `edlist fl freqlist` and click OK.
   It should look, for example, like this:
   
   0  300.19
   3759.74
   8775.99
   8712.96

   Click Cancel to close the list.

4. Enter `stdisp` to start the Shape Tool interface.

5. Proceed as follows:
   + Click , choose Open Shape, select Sinc 1.1000 and click OK
   or
   + Click Shapes’ Classical Shapes’ Sinc and enter:
     
     Size of shape: 1000
     Number of cycles: 1
     
     in the parameter section.

6. Click Manipulate’ Phase modulation acc. to Offset Freq. [manipul offs]

   In the appearing dialog box (see Fig. 2.14):
   • Select Ending at Phase 0.
   • Select Reference = O1 from current Data Set.
   • Check the box Frequencies taken from Frequency List.
   • Set Length of Pulse to 10000.
   • Set Name of Frequency List to freqlist or the name you specified in step 2.
   • Click OK.

7. Check the frequency values in the appearing dialog box (see Fig.
2.15) and click **OK**.

8. Click and click **Save Shape**, enter a **Name** and **Title**, set the **Flip Angle** to 90° and the **Type of Rotation** to excitation.

Now you have created a shape that can be used in a WET experiment.

![Figure 2.14](image-url)
Example 2: Two-step Procedure to a Selective Experiment

1. Read/acquire a 1D reference spectrum.
2. Determine the width of the region to be excited. Let’s assume this is 40 Hz.
3. Create a new dataset.
4. Read the pulse program selgpse.
5. Enter ased and set all acquisition parameters for selgpse, except for the shaped pulse parameters.
6. Enter stdisp to start the Shape Tool interface.
7. Click Shapes ’ Classical Shapes ’ Gauss
8. In the parameter section, enter:
   - **Size of shape**: 1000
   - **Truncation level**: 1
9. Click Options ’ Define Parameter Table
   - Set **Length of shaped pulse** to P12
   - Set **Power level of shaped pulse** to SP2
   - Set **Name of shaped pulse** to SPNAM2
   and click **OK**.
10. Click Analysis ’ Calculate Bandwidth for Refocusing -My
• Enter **Total Rotation**: 180  
  this calculates the bandwidth factor, which is the product of the width of excitation (DeltaOmega) and the pulse length (DeltaT).

• Enter **Delta Omega**: 40 (the bandwidth determined in step 2)  
  this calculates the shaped pulse length DeltaT !!!

• Click *update parameters* to store this pulse length as P12.

11. Click **Analyses’ Integrate Shape**

   • Click *update parameters* to store the name of the shaped pulse to SPNAM2

12. On the associated dataset: enter the command *ased* and check the values of P12 and SP2. If the steps above were performed correctly, they should have the values of DeltaT and Change of power level (+ hard power level), respectively.

Note that if the peak is off-resonance, you need to determine the chemical shift difference between the peak and O1 and set the parameter SPOFFS2 accordingly. To determine the proper value, click 📏, put the cursor on the desired peak and subtract the value of SFO1 from the MHz value displayed in the upper left of the data field.

**Example 3: Create a Shaped Pulse for Sel. Excitation from Integral Regions**

1. Read/acquire a 1D reference spectrum.

2. Click 📊 to switch to integration mode.

3. Click ⬅️ to define regions around the peaks to be excited.

4. Click 📦 to save these regions and return.

5. Create a new dataset.

6. Read the pulse program *selgpse*.

7. Enter *ased* and set all acquisition parameters for *selgpse*, except for the shaped pulse parameters.

8. Switch back the reference spectrum.

9. Enter *stdisp* to start the Shape Tool interface.

10. Click **Shapes’ Classical Shapes’ Gauss**
11. In the parameter section, enter:

**Size of shape**: 1000

**Truncation level**: 1

12. Click *Options ’ Define Parameter Table*
   - Set *Length of shaped pulse* to P12
   - Set *Power level of shaped pulse* to SP2
   - Set *Name of shaped pulse* to SPNAM2

13. Click *Manipulate ’ Calc. Shape from excitation Region* [manipul region]

---

**Figure 2.16**
14. Click **OK**

![Shape.region table]

**Figure 2.17**

15. Click **Options’ Select associated dataset** and select the *selgpse* dataset.

16. Click **update parameters**, specify a name for the shaped pulse and click **OK**.

17. Now switch to the *selgpse* dataset and start the experiment.
Chapter 3
Generating Shapes with st

3.1 Introduction

Syntax

st generate <shape type> <further parameters>

Example

st generate Gauss 1000 1 filename=Gauss.new -nobwcalc

By default, output files are stored under the name <shape type>. If you prefer a different filename, this must be specified with the option:

<filename=newname>

The option <nobwcalc> will prevent automatic calculation of the bandwidth factor: #$SHAPE_BWFAC= 0.0

This example will generate a Gaussian Shape with 1000 points and a truncation level of 1%. The shape is stored as Gauss.new. No automatic calculation of the bandwidth factor is done.
3.2 Basic Shapes

3.2.1 Rectangle Shape

Syntax

\[
\text{st generate Rectangle <size> <amplitude>}
\]

int <size> = shape size in number of points
double <amplitude> = amplitude in %

Example

RF shape: \text{st generate Rectangle 256 100}
Gradient: \text{st generate Rectangle 256 false 100}

3.2.2 Exponential Function (Efunc)

Syntax

\[
\text{st generate Efunc <size> <trunclevel>}
\]

or

\[
\text{st generate Efunc <size> <linebroadening> <sweepwidth>}
\]

int <size> = shape size in number of points
double <trunclevel> = truncation level in %

Conversion: trunclev <-> linebroadening / sweepwidth

\[
\text{trunclev} = \exp (-\beta \times (\text{size}-1)) \times 100
\]
\[
\beta = \frac{(\text{linebroadening} \times \pi)}{(2 \times \text{sweepwidth})}
\]

Example

RF shape: \text{st generate Efunc 256 0.1}
\text{st generate Efunc 256 44 10000}
Gradient: \text{st generate Efunc 256 false 0.1}
\text{st generate Efunc 256 false 44 1000}
3.2.3 Ramp Shape

Syntax

\[
\text{st generate Ramp} \ <\text{size}> \ <\text{start}> \ <\text{end}>
\]

int <size> = shape size in number of points
double <start> = start amplitude in %
double <end> = end amplitude in %

Example

RF shape: \text{st generate Ramp} 256 \ 10 \ 80
Gradient: \text{st generate Ramp} 256 \ false \ 10 \ 80

3.2.4 Quadratic Ramp (QRamp)

Syntax

\[
\text{st generate QRamp} \ <\text{size}> \ <\text{start}> \ <\text{end}>
\]

int <size> = shape size in number of points
double <start> = start amplitude in %
double <end> = end amplitude in %

Example

RF shape: \text{st generate QRamp} 256 \ 10 \ 80
Gradient: \text{st generate QRamp} 256 \ false \ 10 \ 80

3.2.5 Sine Shape (Sinus)

Syntax

\[
\text{st generate Sinus} \ <\text{size}> \ <\text{cycnum}> \ <\text{phase}>
\]

int <size> = shape size in number of points
int <cycnum> = number of cycles to be calculated
double <phase> = phase angle in degree \(0.0 \leq \text{p} \leq 360.0\)

Example

RF shape: \text{st generate Sinus} 256 \ 8 \ 180.0
3.2.6 Trapezoid Shape

Syntax

\[
\text{st generate Trapezoid} \quad <\text{size}> \quad <\text{startAmpl}> \quad <\text{leftLimit}> \quad <\text{centerAmpl}> \quad <\text{rightLimit}> \quad <\text{endAmpl}>
\]

int <size> = shape size in number of points.
double <startAmpl> = Amplitude at Start of Shape (in %).
double <leftLimit> = Left Limit of center region in Points.
double <centerAmpl> = Amplitude at Center of Shape (in %).
double <rightLimit> = Right Limit of center region in Points.
double <endAmpl> = Amplitude at the End of Shape (in %).

Example

RF shape: \text{st generate Trapezoid 1000 0 300 100 700 0}
Gradient: \text{st generate Trapezoid 1000 false 0 300 0 100 700 0}

3.2.7 Triangle Shape

Syntax

\[
\text{st generate Triangle} \quad <\text{size}>
\]

int <size> = shape size in number of points

Example

RF shape: \text{st generate Triangle 256}
Gradient: \text{st generate Triangle 256 false}
3.3 Classical Shapes

3.3.1 Burp Shapes

Syntax

RF shape: \texttt{st generate <shape type> <size>}

int \texttt{<size>} = shape size in number of points

string \texttt{<shape type>} = \{EBurp1 | EBurp2 | IBurp1 | IBurp2 | ReBurp | UBurp1\}

Gradient: \texttt{st generate <shape type> <size> false}

Example

\texttt{st generate IBurp1 256}

Literature

H. Geen & R. Freeman,

3.3.2 Gauss Shape

Syntax

RF shape: \texttt{st generate Gauss <size> <trunclevel>}

Gradient: \texttt{st generate Gauss <size> false <trunclevel>}

int \texttt{<size>} = shape size in number of points
double \texttt{<trunclevel>} = truncation level in %

Example

\texttt{st generate Gauss 256 10}

Literature

C. Bauer, R. Freeman, T. Frenkiel, J. Keeler & A.J. Shaka,
L. Emsley & G. Bodenhausen,
3.3.3 GaussCascade Shapes

Syntax

RF shape: st generate <shape type> <size>

Gradient: st generate <shape type> <size> false

string <shape type> = {GaussCascadeG3 | GaussCascadeG4 | GaussCascadeQ3| GaussCascadeQ5}

int <size> = shape size in number of points

Example

st generate GaussCascadeG4 256

Literature

GaussCascadeG3, GaussCascadeG4:
L. Emsley & G. Bodenhausen,

GaussCascadeQ3, GaussCascadeQ5:
L. Emsley & G. Bodenhausen,

3.3.4 HalfGauss Shape

Syntax

RF shape: st generate HalfGauss <size> <trunclevel>

Gradient: st generate HalfGauss <size> false <trunclevel>

int <size> = shape size in number of points
double <trunclevel> = truncation level in %

Example

st generate HalfGauss 256 10

Literature

J. Friedrich, S. Davies & R. Freeman,
3.3.5 Hermite Shape

Syntax

RF shape: `st generate Hermite <size> <trunclevel> <quadcoeff>`
Gradient: `st generate Hermite <size> false <trunclevel> <quadcoeff>`

int `<size>` = shape size in number of points
double `<trunclevel>` = truncation level in %
double `<quadcoeff>` = coefficient of quadratic term (-4.0 .... +6.0)

Example

`st generate Hermite 256 10 1.0`

Literature

W.S. Warren,

3.3.6 Seduce Shapes

Syntax

`st generate <shape type> <size>`

int `<size>` = shape size in number of points
string `<shape type>` = `{Seduce1 | Seduce3}`

Example

RF shape: `st generate Seduce1 1000`
Gradient: `st generate Seduce1 1000 false`

Literature

M.A. McCoy & L. Mueller,
3.3.7 Sinc Shape

Syntax

\[ \text{st generate Sinc } <\text{size}> <\text{cycnum}> \]

int <size> = shape size in number of points
int <cycnum> = cycnum number of cycles to be calculated

Example

RF shape: \text{st generate Sinc 256 8}
Gradient: \text{st generate Sinc 256 false 8}

Literature

A.J. Temps Jr. & C.F. Brewer,

3.3.8 Sneeze Shape

Syntax

\[ \text{st generate Sneeze } <\text{size}> \]

int <size> = shape size in number of points

Example

RF shape: \text{st generate Sneeze 256}
Gradient: \text{st generate Sneeze 256 false}

Literature

J. M. Nuzillard & R. Freeman,

3.3.9 Snob Shapes

Syntax

\[ \text{st generate } <\text{shape type}> <\text{size}> \]

int <size> = shape size in number of points
string <shape type> = {ESnob | ISnob2 | ISnob3 | RSnob | DSnob}

Example
RF shape: st generate DSnob 256
Gradient: st generate ISnob2 256 false

Literature
E. Kupce, J. Boyd & I.D. Campbell,

3.3.10 Vega Shapes

Syntax
st generate <shape type> <size>
int <size> = shape size in number of points
string <shape type> = {EVega1 | EVega2 | IVega}

Example
RF shape: st generate EVega1 256
Gradient: st generate IVega 256 false

Literature
D. Abramovich & S. Vega,

3.3.11 Shapes defined by Fourier Coefficients (ShapFour)

Syntax
st generate ShapFour <size> <n> <a0> <a1>....<an> <b1>....<bn>
int <size> = shape size in number of points
int <n> = number of coefficients
double <a0> = 0. a-coefficient
double <a1> = 1. a-coefficient
..
double \( a_n \) = n. a-coefficient
double \( b_1 \) = 1. b-coefficient
..  
..  
double \( b_n \) = n. b-coefficient

Example

\texttt{st generate ShapFour 256 4 0.5 -1.147 0.5572 -0.0829 0.0525 0 0 0 0 0}
generates a RSnob shape

### 3.4 Adiabatic Shapes

#### 3.4.1 Hyperbolic Secant Shape (HypSec)

**Syntax**

\texttt{st generate HypSec \langle size\rangle \langle SW\rangle \langle trunclevel\rangle \langle full/half\rangle \langle sweepDir\rangle}

int \langle size\rangle = \text{shape size in number of points}
double \langle SW\rangle = \text{sweep width based on 1sec pulse}
double \langle trunclevel\rangle = \text{truncation level in %}
boolean \langle full / half \rangle = \text{full or half passage (true / false)}
int \langle sweepDir\rangle = \text{Direction of sweep: 1= high to low, -1 = low to high field.}

**Example**

RF shape: \texttt{st generate HypSec 256 6.75 1.0 true -1}

Gradient: \texttt{st generate HypSec 256 false 6.75 1.0 true -1}

**Literature**

M.S. Silver, R.I. Joseph & D.I. Hoult,

#### 3.4.2 SinCos Shape

**Syntax**

RF shape: \texttt{st generate SinCos \langle size\rangle \langle trunclevel\rangle \langle full/half\rangle \langle sweep-}
Generating Shapes with st

**Dir**

Gradient: \texttt{st generate SinCos <size> false <trunclevel> <full/half> <sweepDir>}

- \texttt{int <size> = shape size in number of points}
- \texttt{double <factor> = phase amplitude factor (0.0 <= p <= 16.0)}
- \texttt{boolean <full / half> = full or half passage (true / false)}
- \texttt{int <sweepDir> = Direction of sweep: 1= high to low, -1 = low to high field.}

**Example**

RF shape: \texttt{st generate SinCos 256 8 true -1}

Gradient: \texttt{st generate SinCos 256 false 8 true -1}

**Literature**

M.R. Bendall & D.T. Pegg,

### 3.4.3 SmoothedChirp Shape

**Syntax**

\texttt{st generate SmoothedChirp <size> <SW> <length> <smoothed> <sweepDir>}

- \texttt{int <size> = shape size in number of points}
- \texttt{double <SW> = Total Sweep Width (in Hz)}
- \texttt{double <length> = length of pulse (in usec)}
- \texttt{double <smoothed> = \% to be smoothed}
- \texttt{int <sweepDir> = Direction of sweep: 1= high to low, -1 = low to high field.}

**Example**

RF shape: \texttt{st generate SmoothedChirp 256 40000.0 1500.0 10.0 -1}

Gradients: \texttt{st generate SmoothedChirp 256 false 40000.0 1500.0 10.0 -1}

**Literature**

J. M. Boehlen & G. Bodenhausen,
3.4.4 CompositeSmoothedChirp Shape

Syntax

\[
st \text{ generate CompositeSmoothedChirp} \ \text{<size>} \ \text{<SW>} \ \text{<length>} \\
\text{<smoothed>} \ \text{<sweepDir>}
\]

int <size> = shape size in number of points
double <SW> = Total Sweep Width (in Hz)
double <length> = length of pulse for basic element (in usec)
double <smoothed> = % to be smoothed
int <sweepDir> = Direction of sweep: 1= high to low, -1 = low to high field.

Example

RF shape: \(st \text{ generate CompositeSmoothedChirp} \ 256 \ 40000.0 \ 375.0 \ 10.0 \ -1\)
Gradient: \(st \text{ generate CompositeSmoothedChirp} \ 256 \ false \ 40000.0 \ 375.0 \ 10.0 \ -1\)

Literature

T.L. Hwang, P.C.M van Zijl & M. Garwood

3.4.5 TanhTan Shape

Syntax

\[
st \text{ generate TanhTan} \ \text{<size>} \ \text{<total SW>} \ \text{<length>} \ \text{<zeta>} \ \text{<tan(kappa)>}
\]

int <size> = shape size in number of points
double <total SW> = Total Sweep Width
double <length> = Length of Pulse (in usec)
double <zeta> = Value for zeta.
double <tan(kappa)> = Value for tan(kappa).

Example

RF shape: \(st \text{ generate TanhTan} \ 2000 \ 1000000.0 \ 500.0 \ 10.0 \ 20.0\)
Gradient: \(st \text{ generate TanhTan} \ false \ 2000 \ 1000000.0 \ 500.0 \ 10.0 \ 20.0\)
3.4.6 Wurst Shape

Syntax

\texttt{st generate Wurst <size> <total SW> <length> <power index> <sweep-Dir>}

\begin{itemize}
  \item int <size> = shape size in number of points
  \item double <total SW> = Total Sweep Width
  \item double <length> = Length of Pulse (in usec)
  \item double <power index> = Amplitude Power Index
  \item int <sweepDir> = Direction of sweep: 1= high to low, -1 = low to high field.
\end{itemize}

Example

RF shape: \texttt{st generate Wurst 256 40000.0 1500.0 20.0 -1}

Gradient: \texttt{st generate Wurst 256 false 40000.0 1500.0 20.0 -1}

Literature

E. Kupce & R. Freeman,

3.4.7 Constant-Adiabaticity Shapes

The following constant adiabaticity shapes are implemented:

\begin{itemize}
  \item CaPowHsec (constant adiabaticity hyperbolic secant shape)
  \item CaWurst (constant adiabaticity wurst shape)
  \item CaSmoothedChirp(constant adiabaticity smoothed chirp shape)
  \item CaGauss (constant adiabaticity gauss shape)
  \item CaLorentz (constant adiabaticity lorentz shape)
\end{itemize}
Generating Shapes with st

**Syntax**

\[ \text{st generate <shape type> <size> <additional parameters>} \]

For the additional parameters see the corresponding adiabatic shapes in chapter 3.4. However, for CaGauss and CaLorentz, the additional parameters are:

<SW> <length> <trunclev> <sweepDir>

**Example**

\[ \text{st generate CaWurst 1000 40000.0 1500.0 2.0} \]

**Literature**

A. Tannus & M. Garwood,

---

### 3.5 Special Shapes for Solids Applications

#### 3.5.1 Sines Shape

**Syntax**

\[ \text{st generate Sines <size> <cycnum> <phase> <average> <mod.>} \]

- int \(<\text{size}\) = shape size in number of points
- double \(<\text{cycnum}\) = number of cycles to calculate
- double \(<\text{phase}\) = initial phase angle \((0.0 \leq p \leq 360.0)\)
- double \(<\text{average}\) = average amplitude
- double \(<\text{mod}\) = amplitude of modulation

**Example**

RF shape: \[ \text{st generate Sines 256 2.0 0.0 80.0 10.0} \]

Gradient: \[ \text{st generate Sines 256 false 2.0 0.0 80.0 10.0} \]

**Literature**

S. Hediger, B.H. Meier, R.H. Ernst,
3.5.2 **TangAmplitudeMod Shape**

**Syntax**

\[
\text{st generate TangAmplitudeMod <size> <modulation> <phase> <ampl.>}
\]

- int <size> = shape size in number of points
- double <modulation> = Amplitude of Modulation
- double <phase> = Phase Value (max. 95 degree)
- double <ampl.> = Average Amplitude

**Example**

RF shape: \(\text{st generate TangAmplitudeMod 256 5000.0 26.56 50000.0}\)

Gradient: \(\text{st generate TangAmplitudeMod 256 false 5000.0 26.56 50000.0}\)

**Literature**


3.6 **Special Shapes for Imaging Applications**

3.6.1 **CosSinc Shapes**

**Syntax**

RF shape: \(\text{st generate CosSinc <size> <cycles> <window size>}\)

Gradient: \(\text{st generate CosSinc <size> false <cycles> <window size>}\)

- int <size> = shape size in number of points
- double <cycles> = number of cycles to calculate
- double <window size> = window size in %

**Example**

\(\text{st generate CosSinc 1000 4 50}\)

**Literature**

G.J. Galloway, W.M. Brooks, J.M. Bulsing, I.M. Brereton,
3.6.2 Sine * Sinc Shape (SineSinc)

Syntax

\texttt{st generate SineSinc <size> <cycnum>}

int <size> = shape size in number of points
int <cycnum> = number of cycles to calculate

Example

RF shape: \texttt{st generate SineSinc 256 8}
Gradient: \texttt{st generate SineSinc 256 false 8}

Literature

D.M. Doddrell, J.M. Bulsing, G.J. Galloway, W.M. Brooks, J. Field, M. Irving, & H. Baddeley,

3.7 Special Shapes for Decoupling

3.7.1 Swirl Shapes

Syntax

\texttt{st generate <shape type> <size>}

int <size> = shape size in number of points
string <shape type> = \{Swirl11 | Swirl12 | Swirl17\}

Example

RF shape: \texttt{st generate Swirl11 256}
Gradient: \texttt{st generate Swirl17 256 false}

Literature

H. Geen & J.-M. Boehlen,
Chapter 4

Manipulating Shapes with st

4.1 Introduction

Syntax

st manipulate <shape type> <command> <further parameter>

The following manipulating commands are implemented:

- **offs** - phase modulation according to offset frequencies
- **sinm2** - single sine modulation (+/- offset)
- **cosm2** - single cosine modulation (+/- offset)
- **sweep** - modulation according to linear frequency sweep
- **caSweep** - modulation according to constant adiabaticity frequency sweep
- **power** - calculate power of amplitude
- **scale** - scale the amplitude of a shape
- **addphase** - add constant phase
- **trev** - time reverse a shape
- **expand** - expand shape according to supercycle
4.2 Command offs

Calculates a phase modulation according to offset frequencies.

The manipulate command `offs` has following subcommands:

- `b` - phase modulation beginning at phase 0
- `m` - phase modulation with phase of 0 at middle of shape
- `e` - phase modulation ending at phase 0

**Syntax**

```
st manipulate <shape type> offs b <pulDur> <n> <f1>...<fn>
```

double `<pulDur>` = `pulDur` = length of shaped pulse (in us)
int `<n>` = number of offset frequencies
double `<f1>` = offset frequency 1 (in Hz)
double `<f2>` = offset frequency 2 (in Hz)
.........
.........
double `<fn>` = offset frequency n (in Hz)

**Example**

```
st manipulate Gauss offs b 100 2 2000 3000
```

calculates phase modulation beginning at phase 0.

Optional commands are:

- `f` - frequencies taken from frequency list
- `p` - additional phase setting
- `s` - additional scaling

**Syntax**

```
st manipulate <shape type> offs b f <pulDur> <freqlist>
```

string `<freqlist>` = file with frequency-list

**Example**

```
st manipulate Gauss offs b f 100 freqlist
```
Syntax

```
st manipulate <shape type> offs m p <pulDur> <n> <f1> <p1>...<fn> <pn>
```

- `double <pulDur>` = `pulDur` = length of shaped pulse (in us)
- `int <n>` = number of offset frequencies
- `double <f1>` = offset frequency 1 (in Hz)
- `double <p1>` = initial phase 1 (in degree)
- `double <f2>` = offset frequency 2 (in Hz)
- `double <p2>` = initial phase 2 (in degree)
- `........`
- `........`
- `double <fn>` = offset frequency n (in Hz)
- `double <pn>` = initial phase n (in degree)

Example

```
st manipulate Gauss offs m p 100 2 2000 90 3000 300
```

Syntax

```
st manipulate <shape type> offs e s <pulDur> <n> <f1> <s1> <f2> <s2>
```

- `double <pulDur>` = `pulDur` = length of shaped pulse (in us)
- `int <n>` = number of offset frequencies
- `double <f1>` = offset frequency 1 (in Hz)
- `double <s1>` = scaling factor 1 (in %)
- `double <f2>` = offset frequency 2 (in Hz)
- `double <s2>` = scaling factor 2 (in %)
- `........`
- `........`
- `double <fn>` = offset frequency n (in Hz)
- `double <sn>` = scaling factor n (in %)

Example

```
st manipulate Gauss offs e s 100 2 2000 50 3000 75
```

Note that multiple optional commands can be used, for example:

```
st manipulate Gauss offs m f s 100 50 75 freqlist
```
```
st manipulate Gauss offs e f p 100 90 180 freqlist
```
4.3 Command sinm2

Calculates an amplitude modulation such that the pulse excites at two symmetric sidebands (+/- offset) with opposite phase.

**Syntax**

```
st manipulate <shape type> sinm2 <pulDur> <offset>
```

double <pulDur> = length of shaped pulse (in us)

double <offset> = offset frequency (in Hz)

**Example**

```
st manipulate Gauss sinm2 1000 3000
```

4.4 Command cosm2

Calculates an amplitude modulation such that the pulse excites at two symmetric sidebands (+/- offset) with the same phase.

**Syntax**

```
st manipulate <shape type> cosm2 <pulDur> <offset>
```

double <pulDur> = pulDur = length of shaped pulse (in us)

double <offset> = offset frequency (in Hz)

**Example**

```
st manipulate Gauss cosm2 1000 3000
```

4.5 Modulation according to Frequency Sweep

4.5.1 Command sweep

Calculates a phase modulation according to a linear frequency sweep.

**Syntax**

```
st manipulate <shape type> sweep <pulDur> <sw>
```
Manipulating Shapes with st

```plaintext
double <pulDur> = length of shaped pulse (in us)
double <sw> = total sweep-width (in Hz)
```

**Example**
```
st manipulate Gauss sweep 1000 5000
```

### 4.5.2 Command caSweep

Calculates a phase modulation according to a constant adiabaticity frequency sweep: \(\text{phase}(t) = \int \text{frequency}(t) = \int \int \text{amplitude}(t)\)

**Syntax**
```
st manipulate <shape type> caSweep <pulDur> <sw>
```
```
double <pulDur> = length of shaped pulse (in us)
double <sw> = total sweep-width (in Hz)
```

**Example**
```
st manipulate Gauss caSweep 1000 5000
```

### 4.6 Command power

Calculate the power of an amplitude. This allows you to compare a theoretical shape with an observed shape on the scope.

**Syntax**
```
st manipulate <shape type> power <exponent>
```
```
double <exponent> = exponential factor
```

**Example**
```
st manipulate Gauss power 2
```

### 4.7 Command scale

Scale the amplitude of a shape to a given percentage.
4.8 Command addphase

Adds a constant phase to a shape.

**Syntax**

```
st manipulate <shape type> addphase <phase>
```

double <phase> = phase constant to be added in degree

**Example**

```
st manipulate Gauss 90.0
```

4.9 Command trev

Time reverse a shape. Since shapes are often optimized for a particular rotation (e.g. Iz → ly, excitation), reversing the rotation (ly → Iz, flipback) will only work reliably if the shape is time reversed as well.

**Syntax**

```
trev
```

*trev* needs no additional parameters.

**Example**

```
st manipulate HalfGauss trev
```
4.10 Command expand

Expand shape according to a phase list (supercycle).

Syntax

```
  st manipulate <shape type> expand <n> <phase1> ... <phasen>
```

- `int <n>` = number of phases to expand
- `double <phase1>` = first phase value
- ...
- `double <phasen>` = nth phase value

Example

```
  st manipulate Gauss expand 4 0 0 180 180
```
Chapter 5

Analyzing Shapes with st

5.1 Introduction

Syntax

```
st analyze <shape type> <analyze command> <further parameter>
```

The following analyzing commands are available:

- **bandw2** calculate bandwidth for excitation \( \sqrt{Mx^2 + My^2} \)
- **bandw2i** calculate bandwidth for inversion \(-Mz\)
- **bandw2ry** calculate bandwidth for refocusing \(-My\)
- Special Bandwidth Calculations:
  - **bandw2e** calculate bandwidth for excitation \(-My\)
  - **bandw2r** calculate bandwidth for refocusing \(My \cdot \sqrt{Mx^2 + My^2}\)
  - **bandw2rx** calculate bandwidth for refocusing \(+Mx\)
- **calcb1mo** calculate \(\gamma B1(max)/2\pi\) at offset
- **calcb1adia** calculate \(\gamma B1(max)/2\pi\) for adiabatic shapes
- **bsiegert3** - calculate Bloch-Siegert shift
- **calcpav** calculate average power level
• integr3 - integrate shape and calculate power level compared to hard pulse
• integradia - integrate adiabatic shapes and calculate power level compared to hard pulse.

The following commands are only available for the st analyze command. In the stdisp graphical interface, they have been replaced (see chapter 5.12).
• integr integrate shape and compare to square.
• integr2 calculate power level compare to hard pulse.
• calcb1m calculate $\gamma B1(\text{max})/\pi$ on resonance.

### 5.2 Command bandw2

Calculate the bandwidth for excitation ($\sqrt{(Mx^2+My^2)}$).

**Syntax**

```
st analyze Gauss bandw2 <rotation>
```

**Example**

```
st analyze Gauss bandw2 90
```

Result: bandwidth factor $\Delta \omega \times \Delta T$.

### 5.3 Command bandw2i

Calculates the bandwidth for inversion (-Mz).

**Syntax**

```
st analyze Gauss bandw2i <rotation>
```

**Example**

```
st analyze Gauss bandw2i 180
```
Result: bandwidth factor $\Delta \omega \ast \Delta T$.

5.4 Command bandw2ry

Calculates bandwidth for refocusing (-My).

**Syntax**

```bash
st analyze Gauss bandw2ry <rotation>
```

double <rotation> = total rotation (in degree)

**Example**

```bash
st analyze Gauss bandw2ry 180
```

Result: bandwidth factor $\Delta \omega \ast \Delta T$.

5.5 Special Bandwidth Calculations

- **bandw2e**: calculate bandwidth for excitation (-My)
- **bandw2r**: calculate bandwidth for refocusing (My $\sqrt{(Mx2+My2)}$
- **bandw2rx**: calculate bandwidth for refocusing (+Mx)

Syntax and Result of these commands are identical to the bandwidth calculation commands described above.

5.6 Command calcb1mo

Calculates $\gamma B1(max)/\pi$ at offset.

**Syntax**

```bash
st analyze Gauss calcb1mo <pulDurShape> <rotation> <offset>
```

double <pulDurShape> = length of shaped pulse (in us)
double <rotation> = total rotation (in degree)
double <offset> = offset (in Hz)
Example

*st analyze Gauss calcb1mo 100 90 3000*

Result:
- maximum gB1/2π (on resonance)
- corresponding 90 degree square pulse

### 5.7 Command calcb1adia

Calculates $\gamma B1(max)/2\pi /\sqrt(Q)$ for adiabatic shapes.

**Syntax**

*st analyze SmoothedChirp calcb1adia <pulDurShape>*

double <pulDurShape> = length of shaped pulse (in us)

Example

*st analyze SmoothedChirp calcb1adia 10000*

Result:
- sweep rate on resonance (in Hz/sec)
- maximum $\gamma B1/2\pi /\sqrt(Q)$ (on resonance)

### 5.8 Command bsiegert3

Calculates the Bloch-Siegert shift.

**Syntax**

*st analyze <shape type> bsiegert3 <pulDur> <γB1> <offset>*

double <pulDur> = length of shaped pulse (in us)
double <γB1> = $γB1(max)$ (in Hz)
double <offset> = offset relative to frequency of RF-pulse (in Hz)

Result:
- phase difference due to Bloch-Siegert shift (in degree)
5.9 Command calcpav

Calculates the average power.

calcpav needs no additional parameters.

Example

```
st analyze Gauss calcpav
```

Result:

- amplitude relative to a square pulse of 100%
- corresponding difference in dB.
- power relative to a square pulse of 100%

5.10 Command integr3

Integrates shape and calculates power level compared to hard pulse.

Syntax

```
st analyze Gauss integr3 <pulDurShape> <pulDurHard> <totRot>
```

double <pulDurShape> = length of shaped pulse (in us)
double <pulDurHard> = length of reference hard pulse (in us)
double <totRot> = total rotation (in degree).

Example

```
st analyze Gauss integr3 10000 6.6 180
```

Result:

- integral ratio compared to a square pulse of 100% (Flip angle not evaluated)
- corresponding difference in dB. (Flip angle not evaluated)
- change of power level compared to level of hard pulse in dB
5.11 Command integradia

Integrates adiabatic shapes and calculates power level compared to hard pulse.

**Syntax**

```st analyze <adiabatic shape> integradia <pulDurShape> <pulDurHard>

double <pulDurShape> = length of shaped pulse (in us)
double <pulDurHard> = length of reference hard pulse (in us)
```

**Example**

```st analyze SmoothedChirp integradia 10000 8```

Result:

- sweep rate on resonance (in Hz/sec)
- $\gamma B_1(\text{max})/2\pi / \sqrt(Q)$

5.12 Routines which have been replaced in stdisp

The commands `integr` and `integr2` are available for `st`, but have been replaced by `analyze integr3` in `stdisp`. The command `calcb1m` for `st` has been replaced by `analyze calcb1mo` with offset = 0 in `stdisp`

5.12.1 Command integr

Integrates shape and compare to square.

`integr` needs no additional parameters

**Example**

```st analyze Gauss integr```

Result:

- integral ratio compared to a square pulse of 100%
- corresponding difference in dB
5.12.2 Command integr2

Calculates the power level compared to a hard pulse.

**Syntax**

```
st analyze <shape type> integr2 <pulDurShape> <pulDurHard>
```

double <pulDurShape> = length of shaped pulse (in us)
double <pulDurHard> = length of reference hard pulse (in us) of same flip angle

**Example**

```
st analyze Gauss integr3 10000 6.6 180
```

Result:

- integral ratio compared to a square pulse of 100%
- corresponding difference in dB.
- change of power level compared to level of hard pulse in dB.

5.12.3 Command calcb1m

Calculates $\gamma B1(\text{max})/2\pi$ on resonance.

**Syntax**

```
st analyze Gauss bandw2i <rotation>
```

double <pulDurShape> = length of shaped pulse (in us)
double <rotation> = total rotation (in degree)

**Example**

```
st analyze Gauss calcb1m 100 180
```

Result:

- maximum $\gamma B1$ (on resonance).
- corresponding 90 degree square pulse
Chapter 6
Miscellaneous st functions

6.1 Introduction

• **add** - add shapes (up to 10).
• **convert, convertgr** - convert shapes/gradients from conventional ASCII format to JCAMP-DX format.

6.2 Commands convert, convertgr

This commands convert shapes/gradients from conventional ASCII format to JCAMP-DX format.

Syntax

```
    st convert <input> <output> <type> <#freq> [<deltaOmega*deltaT>]
```

string <type> = types are: Universal, Excitation, Inversion, etc.
string <input> = input filename
string <output> = output filename
int <#freq> = number of off resonance frequencies in shape
double <deltaOmega*deltaT> = value for deltaOmega*deltaT (this parameter is optional)
Examples

st convert Gauss.ascii Gauss.jcamp Excitation 0
st convertgr Gauss.ascii Gauss.jcamp

Required format of the ascii input files:

<table>
<thead>
<tr>
<th>RF-shapes</th>
<th>Gradients</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFVERSION_F</td>
<td>GRADIENT</td>
</tr>
<tr>
<td>-5.14574 0.0000</td>
<td>1.000000e-02</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>-5.16086 0.0000</td>
<td>1.000000e-02</td>
</tr>
</tbody>
</table>

6.3 Command add

Add existing shapes to form a new shape.

Syntax

st add <alignment> <n> <input1> <scale1> ... <inputn> <scalen> <result>

int <alignment> = with respect to beginning = 0, center = 1, end = 2 of shape
int <n> = number of shapes to be added (max. 10 shapes are allowed)
string <input1> = filename of first shape to be added
double <scale1> = scaling of first shape
...
...
string <inputn> = filename of nth shape to be added
double <scalen> = scaling of nth shape
string <result> = result of the addition

Example

st add 0 2 Gauss.1 100 Gauss.2 50 Gauss.result
Chapter 7
Shapes in AU programs

7.1 Using Shape Tool commands in AU Programs

Here you find an example of an AU program using Shape Tool commands.

```c
#include <stdio.h>
#include <stdlib.h>
#include <strings.h>
#include <limits.h>
#include <ShapeIO/ShapeIOC.h>

#define MAX_ANZ 20
char result[PATH_MAX];
char filename[PATH_MAX];
char *singleResult;
int i, numOfParams;
double retValue[MAX_ANZ];
double bandWidthFactor;

/* read gaussian shape and call analyze command bandw2 */

(void)sprintf(filename, "%slists/wave/Gauss", getstan(0, 0));
```
numbOfParams = analyzeShapeC(filename, "bandw2", "90", &result[0]);
if (numbOfParams <= MAX_ANZ)
{
    i = 0;
    singleResult = strtok(result, ","); /* scan result string for results */
    retValue[i] = atof(singleResult);
    while (((singleResult = strtok(0, ", "))))
    {
        retValue[++i] = atof(singleResult);
        if (i >= numbOfParams) break;
    }
    bandWidthFactor = retValue[0];
}
/* do further calculations with bandWidthFactor */
........
........
QUIT

In this example the command `analyzeShapeC(...)` reads a gaussian shape and applies the command `bandw2` with the parameter total rotation = 90° and stores the result of the calculation into the char [] result. The return value of `analyzeShapeC` is the number of results calculated. In the case of `bandw2` only one result, the bandWidthFactor $\Delta\omega \ast \Delta T$ is returned. Now the Bandwidth factor can be used for further calculations in the AU program.

Most of the analyze commands return more than one result. These are then separated by semicolons in the returned char [] result. The while loop in the example scans this array and copies the single results to the double array retValue. Now all results of the analyze command are available for further calculations.
Chapter 8
Shape File Format

8.1 Format of a Gaussian Shape File

```plaintext
##TITLE= /tshome/exp/stan/nmr/lists/wave/Gauss
##JCAMP-DX= 5.00 Bruker JCAMP library
##DATA TYPE= Shape Data
##ORIGIN= Bruker Analytik GmbH
##OWNER= <wk>
##DATE= 00/03/23
##TIME= 08:19:08
##$SHAPE_PARAMETERS= Type: Gauss ; Truncation Level: 1
##MINX= 1.000000e+00
##MAXX= 9.999954e+01
##MINY= 0.000000e+00
##MAXY= 0.000000e+00
##$SHAPE_EXMODE= Universal
##$SHAPE_TYPE=
##$SHAPE_USER_DEF=
##$SHAPE_REPHFAC=
##$SHAPE_TOTROT= 9.000000e+01
##$SHAPE_BWFAC= 2.122000e+00
##$BWFAC50 =
```
## $SHAPE_INTEGFAC = 4.115776e-01
## $SHAPE_MODE = 0
## NPOINTS = 1000
## XYPOINTS = (XY..XY)
  1.000000e+00, 0.000000e+00
  1.018591e+00, 0.000000e+00
  1.037490e+00, 0.000000e+00
  1.056700e+00, 0.000000e+00
  1.076227e+00, 0.000000e+00
  1.096073e+00, 0.000000e+00
  1.116245e+00, 0.000000e+00
  1.136746e+00, 0.000000e+00
  1.157580e+00, 0.000000e+00
  1.178753e+00, 0.000000e+00
  1.200269e+00, 0.000000e+00
  ...
  ...
  ...
  1.200269e+00, 0.000000e+00
  1.178753e+00, 0.000000e+00
  1.157580e+00, 0.000000e+00
  1.136746e+00, 0.000000e+00
  1.116245e+00, 0.000000e+00
  1.096073e+00, 0.000000e+00
  1.076227e+00, 0.000000e+00
  1.056700e+00, 0.000000e+00
  1.037490e+00, 0.000000e+00
  1.018591e+00, 0.000000e+00
  1.000000e+00, 0.000000e+00
## END=