VARIFORC User Manual
Chapter 5:

Plot FORC diagrams

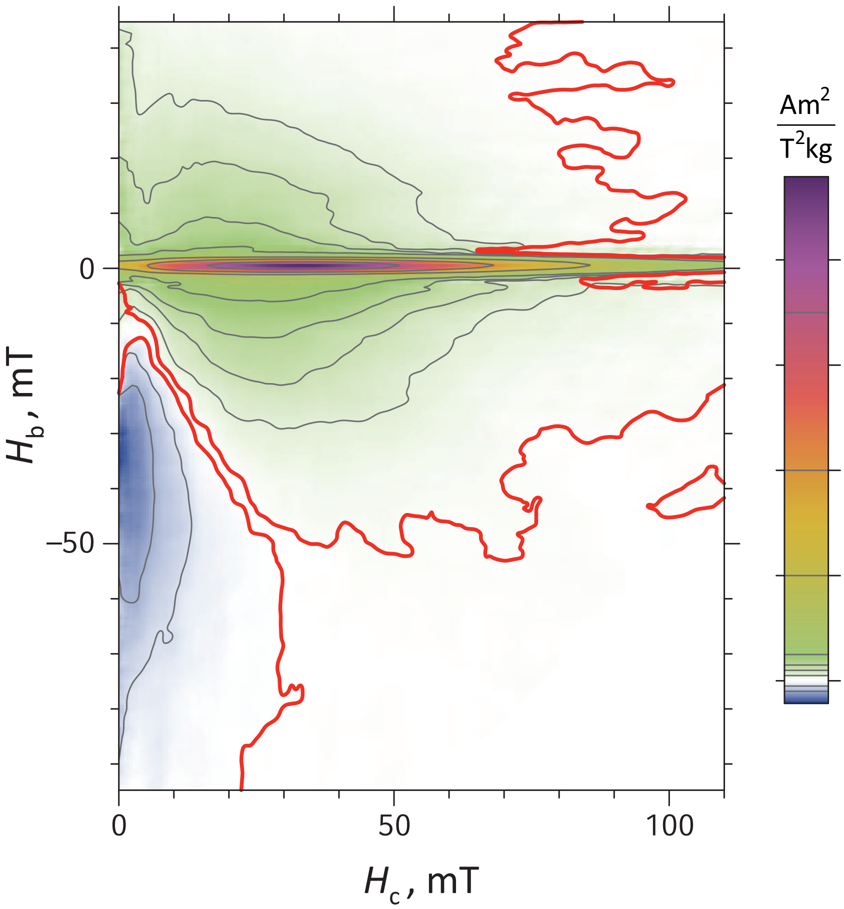


Table of contents

5.1 PlotFORC highlights

`PlotFORC` is a module of the VARIFORC package that is used for plotting FORC diagrams generated by `CalculateFORC` and other VARIFORC modules. Although FORC diagrams are displayed by these modules as part of their standard output, advanced plotting options are only available with `PlotFORC`. These options enable the customization of FORC diagrams for best presentation results and professional printing.

Like other VARIFORC modules, `PlotFORC` is controlled by user-defined options stored in a special parameter file that can be used for batch processing. `PlotFORC` offers the following advanced plotting options (Fig. 5.1), with unique VARIFORC features highlighted in *cursive*.

1) Choose arbitrary plotting ranges, *including diagonal limits over the upper and lower quadrants, in conformity with the intrinsic range of FORC functions*.
2) *Use appropriated FORC units, as part of the VARIFORC unit management system.*
3) *Choose among a palette of color scales specially developed for best FORC diagram representation.*
4) *Customize color scales to enhance weak amplitudes and clip large values.*
5) Customize diagram axes and color scale bar.
6) *Use arbitrary aspect ratios for plotting details of horizontal and vertical ridges.*
7) Add a given number of contour lines or *draw contour lines at specified levels. Exclude short contour loops generated by noise.*
8) Export the FORC diagram in various vector and raster graphical formats.

Use this manual to learn about `PlotFORC`, and see the quick VARIFORC guide for a short summary and option reference.

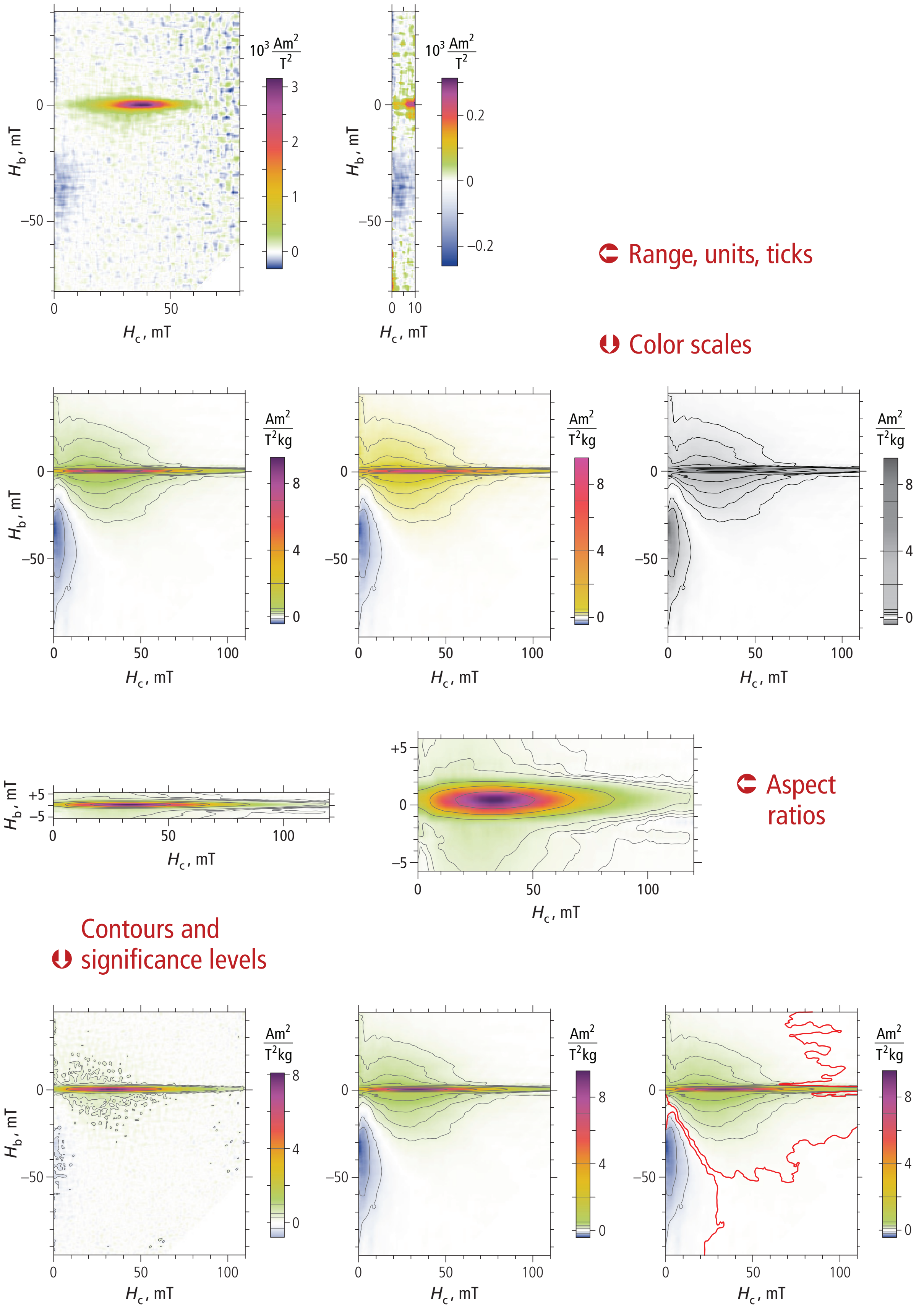


Fig. 5.1: Graphical summary of advanced plotting options offered by `PlotFORC`.

5.2 Using PlotFORC

Like all VARIFORC modules, `PlotFORC` runs on a Wolfram Mathematica® or PlayerPro® notebook. A starting copy of this notebook is provided with the VARIFORC installation package under:

```
VARIFORC_Install/Modules/PlotFORC/VARIFORC_PlotFORC.cdf
```

You can copy this file to a different folder for your convenience. The notebook stores all processing steps and graphical results; therefore, it is best renamed and saved with reference to the processed FORC data. The default notebook provided with the installation package is a .cdf file (computable document format) that works with both Wolfram Mathematica® and PlayerPro®. The content of a .cdf notebook cannot be modified with Player Pro®, but existing commands can be evaluated and results saved without restrictions. The .cdf notebook can be saved as a regular notebook (.nb) with Wolfram Mathematica®.

All `PlotFORC` notebooks begin with the following command line:

```
Get[FileNameJoin[{$HomeDirectory,"VARIFORC","VARIFORC_PlotFORC_code.txt"}]]
```

which uploads `PlotFORC` to the computation kernel. This line, as any other command in Mathematica® notebooks, is executed by placing the cursor on it and pressing the keys SHIFT and ENTER at the same time. A copyright message will appear below this line, confirming that `PlotFORC` has been uploaded successfully.

`PlotFORC` is then called by the command line

```
VARIFORC`PlotFORC
```

to be executed by pressing the keys SHIFT and ENTER at the same time. At this point, `PlotFORC` starts a system dialog for:

1) uploading a parameter file that contains user-defined processing options,
2) uploading a FORC matrix produced by other VARIFORC modules, and
3) defining an output file where the FORC diagram is exported as graphics.

The notebook appearance upon performing these steps is shown in Fig. 5.2.

The files required by steps 1) and 2) should be ready before `PlotFORC` is started. The parameter file is an unformatted text file that contains all processing options to be used by `PlotFORC`. You can find a template of this file with universally valid standard options in the VARIFORC installation package under:

```
VARIFORC_Install/Modules/PlotFORC/Default_VARIFORC_PlotFORC_Parameters.txt
```

You may copy this file to a directory hosting all VARIFORC processing files related to given FORC measurements. Parameters in this file can be modified with a text editor, according to your processing requirements (see section 5.4 and the VARIFORC quick guide). File upload and output file naming are explained in section 5.3.

❶ Upload source codes

```
In[1]:= Get[FileNameJoin[{$HomeDirectory, "VARIFORC", "VARIFORC_PlotFORC_code.txt"}]]

        Function VARIFORC`PlotFORC for plotting FORC measurements.
        [VARIFORC package v1.0 for Wolfram Mathematica and Mathematica Player Pro.
         © 2014 by Ramon Egli. All rights reserved.]
```

If used for scientific publications and presentations please cite as follows:
Egli, R. (2013). VARIFORC: An optimized protocol for calculating non-regular first-order reversal curve (FORC) diagrams. *Global and Planetary change 110*, 302-320.
http://dx.doi.org/10.1016/j.gloplacha.2013.08.003

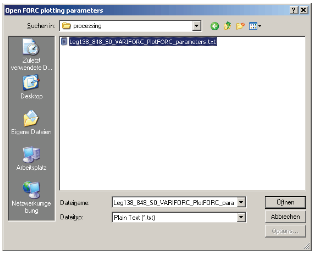
❷ Call PlotFORC

```
In[2]:= VARIFORC`PlotFORC
```

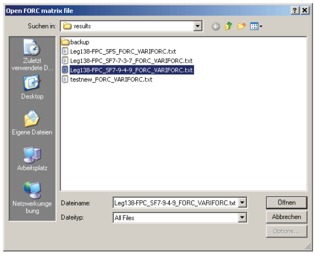
❸ Read auxiliary files

```
        Initialization...
```

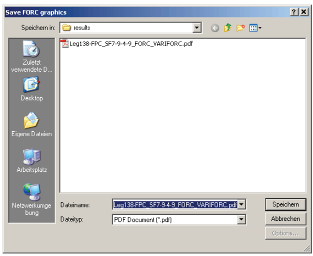
❹ Upload parameter file



❺ Upload FORC matrix file



❻ Define output graphics file



❼ Begin processing

```
Read parameter file...

Input parameters from C:/.../Example/CalculateFORC_Parameters.txt:
INPUT 01. Plotted Hc-range ....................; All
INPUT 02. Plotted Hb-range ....................; All
INPUT 03. Diagonal FORC space limits ..........; None
INPUT 04. FORC unit factor ....................; Automatic
INPUT 05. Color scale range ...................; Automatic
INPUT 06. Color function specifications .......; Iridescent
INPUT 07. Expanded color range option .........; Automatic
INPUT 08. Field ticks specifications ..........; 50,5
INPUT 09. Vertical exaggeration factor ........; 1
INPUT 10. Color bar vertical stretch ..........; 1
INPUT 11. Contour specifications ..............; -100,100,200,500,2000,4000,7000
INPUT 12. Contour spline type .................; BSpline, 100
INPUT 13. Exclusion limit for short contours ..; 0.1
INPUT 14. Contour color .......................; gray
INPUT 15. Significance contour specifications .; 3
INPUT 16. Significance contour color ..........; red
INPUT 17. FORC plot export format .............; PDF
```

Fig. 5.2: Initialization of the `PlotFORC` notebook.

After completing file upload and saving dialogs, `PlotFORC` proceeds autonomously until the end, without requesting any further action by the user. The processing status is continuously updated by progress messages, e.g.

```
Prepare color scale ...
```

warning messages, e.g.

```
WARNING: Minimum FORC values are positive. The default color scale will be used.
```

error messages, e.g.

```
Unknown contour color. Program aborted. [INPUT 14]
```

summary tables, e.g.

```
Range parameters for color scale Iridescent in original units:
FORC range .........: from -429.391 ±100.143 to 9580.5 ±210.068
Color scale range ..: from -429.391          to 9580.5
Green at ...........: 429.391
```

and the FORC diagram plot. `PlotFORC` ends with the following message containing the total computation time:

```
PROGRAM END. Total computation time 1m 6s.
```

You can save the notebook file with all messages and results shown above for your later records and re-use it.

`PlotFORC` error messages usually produce a program abort. If one of these messages appears, an error was encountered either in the parameter file, e.g.:

```
Contour spline type is not “Bezier” or “BSpline” followed by a polynomial degree.
Program aborted [INPUT 12].
```

or in the matrix file, e.g.:

```
Error encountered in reading FORC matrix file (Data matrix).
   Valid files must be produced by a VARIFORC function! Program aborted.
```

FORC matrix files produced by VARIFORC modules are fully compatible and do not generate reading errors unless accidentally modified. Error messages contain hints about the encountered problem. Errors generated by the parameter file arise from incorrect option specifications (see section 5.5 for acceptable options), or by altered formats. In the latter case, generate a new parameter file from the original copy provided with the installation package.

- Large parts of the `PlotFORC` code are dedicated to error handling in order to avoid unexpected crashes or incorrect results. Although VARIFORC modules have been extensively tested, the occurrence of unexpected errors cannot be completely excluded. Such errors might generate a cascade of other errors and/or a program crash.
- Unexpected errors likely cause Mathematica® notebooks to freeze. In this case, you should forcefully terminate the Mathematica® Kernel as described in Chapter 2.

- All information concerning `PlotFORC` runs is stored in the Mathematica® notebook. You can save the notebook with its content for your records.
- Use only one Wolfram Mathematica at the time.
- In order to avoid excessive memory usage, process a single dataset in each notebook.

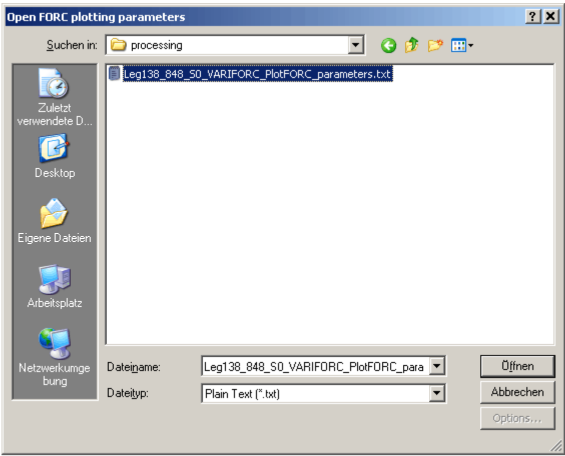
5.3 PlotFORC file management

VARIFORC has a modular structure, so that each module performs a specific operation by reading data stored in measurement files or files containing processed data, and export results to one or more files that can be used by other VARIFORC modules. `PlotFORC` is the last stage of FORC processing. It reads a FORC matrix file and the associated FORC error file where FORC function values and corresponding standard errors are stored.

Results, in form of high-quality FORC diagrams, are plotted in the Mathematica® notebook and exported as graphics. `PlotFORC` calls the file dialog of your operating system in order to let you upload and export files, as described in the following.

5.3.1 Parameter file upload

The parameter file is uploaded through the following dialog window:



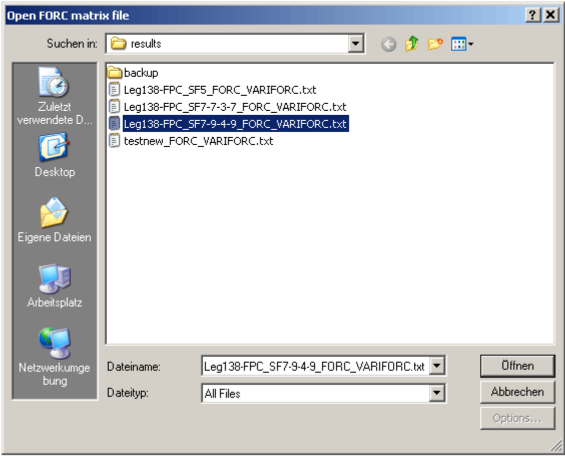
whose detailed appearance depends on your operating system. `PlotFORC` automatically selects file names ending with `VARIFORC_PlotFORC_parameters.txt`, which is default for all `PlotFORC` parameter files. It is strongly recommended to keep this file name ending, so to avoid confusion with parameter files of other VARIFORC modules. The first part of the file name can be related to the sample being processed, e.g.:

```
/.../Sample01_VARIFORC_PlotFORC_parameters.txt
```

If you do not find the parameter file in the expected directory, remove file name and file type filters, so that all files will be displayed.

4.3.2 FORC matrix upload

The FORC matrix (previously generated by other VARIFORC modules) is uploaded through the following dialog window:

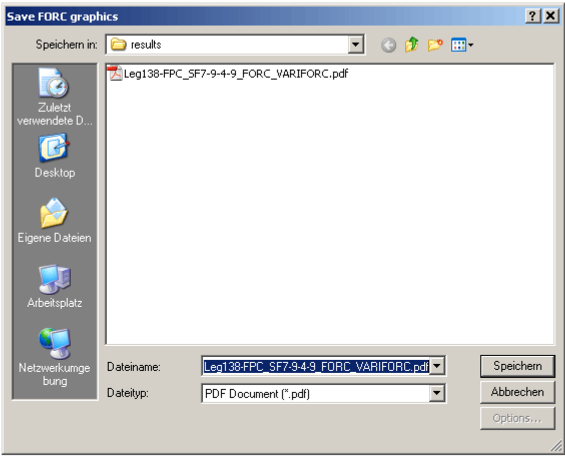


whose detailed appearance depends on your operating system. Be careful to choose a single FORC matrix file. Only file names ending with `FORC_VARIFORC.txt` and `FORCStandardError_VARIFORC.txt` are shown, according to the naming system used for FORC matrices. Depending on plotting options specified in the parameter file, `PlotFORC` might need to upload the error matrix related to the FORC data. This matrix is stored in a file with same name root, ending with `FORCStandardError_VARIFORC.txt`. `PlotFORC` looks automatically for error matrix files if necessary, without further input requests.

- FORC matrix files (ending with `FORC_VARIFORC.txt`), and corresponding FORC error matrix files (ending with `FORCStandardError_VARIFORC.txt`) should be stored always in the same folder, as when produced by a VARIFORC module.
- VARIFORC modules such as `PlotFORC` automatically look for FORC error matrix files if needed, assuming that these files are stored in the same folder of the corresponding FORC matrix.

5.3.3 FORC diagram export

If a graphics format for exporting the FORC diagram plot has been specified in the parameter file, a file name root will be asked through the following system dialog window:



whose detailed appearance depends on your operating system. `PlotFORC` opens by default the same directory of the FORC matrix file and suggests a name root that is extracted from it.

- Default output file names suggested by `PlotFORC` do not contain any reference to the chosen plotting parameters.
- If you want to store several versions of the same FORC diagram (e.g. with and without contours), add a hint in the file name of the stored graphics.

5.4 Editing the PlotFORC options

`PlotFORC` is controlled by 17 parameters that are uploaded from a parameter file ending with `_VARIFORC_PlotFORC_Parameters.txt`. This is an editable text file with a template provided with the installation package (see section 5.2). This template contains universal parameters that can be used with practically any type of FORC diagram. You can copy this file to any folder, typically the same folder containing the FORC matrix and/or the `PlotFORC` Mathematica® notebook, and use it as is. Nevertheless, in order to exploit all `PlotFORC` capabilities and obtain best results, you should adapt these parameters to your requirements. For this purpose, open the `PlotFORC` parameter file with a text editor. You will see a table similar to the following example:

```
Input parameters for package VARIFORC_PlotFORC; (version 2.04).

INPUT 01. Plotted Hc-range .....................; All
INPUT 02. Plotted Hb-range .....................; All
INPUT 03. Diagonal FORC space limits ...........; None
INPUT 04. FORC unit factor .....................; Automatic
INPUT 05. Color scale range ....................; Automatic
INPUT 06. Color function specifications ........; Iridescent
INPUT 07. Expanded color range option ..........; Automatic
INPUT 08. FORC diagram ticks specification .....; 0.05,5
INPUT 09. Vertical exaggeration factor .........; 1
INPUT 10. Color bar range ......................; Automatic
INPUT 11. Color bar normalization ..............; Yes
INPUT 12. Color bar ticks ......................; Contours
INPUT 13. Color bar vertical stretch ...........; 1
INPUT 14. Contour level specification ..........; -10,10,20,50,80%
INPUT 15. Contour spline type ..................; BSpline, 100
INPUT 16. Exclusion limit for short contours ...; 0.01
INPUT 17. Contour style ........................; gray
INPUT 18. Significance contour specification ...; 3, All
INPUT 19. Significance contour color ...........; red
INPUT 20. FORC diagram export format ...........; PDF
INPUT 21. Vertical profile normalization .......; No
```

Each line of this table (except for the first one) consists of a parameter/option description (e.g. `INPUT 01. Plotted Hc-range`) followed by the corresponding parameter value (e.g. `All`). Parameters are separated from their descriptions by a semicolon (;). Multiple parameters in the same row (e.g. a list of numbers) are always separated by a colon (,) and spaces have no meaning. You can change these parameters according to the guidelines given in section 5.5 and save the parameter file with an appropriated name related to its usage context.

`INPUT 18` is a new option available with version 2 of `PlotFORC`.

<u>Example 1:</u> If you routinely process FORC diagrams obtained from high-resolution measurements up to 120 mT, you can store the parameter file as `highres_120mT_VARIFORC_PlotFORC_Parameters.txt`, and recall it every time a final diagram is prepared with `PlotFORC`.

<u>Example 2:</u> Plotting parameters to be used only with the FORC matrix `File_1_FORC_VARIFORC.txt` can be stored in a parameter file called `File_1_VARIFORC_PlotFORC_Parameters.txt` for later records.

- Always save the processing parameters as <u>unformatted text files</u>. If you use a text processor such as Microsoft Word, do not forget to save as text (`.txt`) only. Formatted texts are not recognized by `PlotFORC`.
- <u>Do not change the table structure</u>; in particular, do not add new lines. Multiple parameters representing the same input (e.g. `INPUT 11 contour specifications`) must be entered as a single line.
- <u>Acceptable parameter formats</u> are explained in detail in section 5.5. Incorrect or unrecognized formats generate error messages referring to the corresponding input line (e.g. `INPUT 01`).

The following suggestions help you with an efficient management of plotting parameters:

- Several input parameters can be set to automatic options, letting `PlotFORC` <u>choose optimal values</u>. Use automatic options if possible, unless you have specific processing requirements. For your convenience, the parameter file provided with the installation package is already based on automatic options whenever possible.
- <u>Use consistent FORC protocols</u> for your measurements, so that only few sets of processing parameters are required. For example, all magnetofossil-bearing sediments can be measured with the same protocol and plotted over the same FORC range.
- <u>If you have generated an invalid parameter file</u> and you do not know how to restore the proper format, create a new file starting from the template provided with the installation package. For this reason, <u>never overwrite templates</u> in the installation package.

5.5. Full references to PlotFORC options

All `PlotFORC` parameter options are described in this section. You can refer to the VARIFORC quick guide for a compact summary.

Table of contents:

INPUT 01-02. Plotted range

The rectangular FORC space over which the FORC diagram is plotted can be chosen by entering the corresponding H_c- and H_b-intervals. The following settings:

```
INPUT 01. Plotted Hc-range ....; All
INPUT 02. Plotted Hb-range ....; All
```

are used for plotting the whole FORC matrix. Often, the range covered by significant FORC amplitudes is smaller, as seen on plots of the signal-to-noise-ratios generated during FORC calculation (e.g. Fig. 5.2a). In this case, the plotted FORC space is conveniently limited by entering explicit H_c- and H_b-ranges with `INPUT 01` and `INPUT 02`, respectively. For example,

```
INPUT 01. Plotted Hc-range ....; 0, 110
INPUT 02. Plotted Hb-range ....; -95, 45
```

is used for plotting the FORC diagram shown in Fig. 5.3b over H_c- and H_b-ranges of 0-100 mT and −95-45 mT, respectively. You can also mix the options for `INPUT 01` and `INPUT 02`, using `All` for one FORC coordinate, and an explicit range for the other.

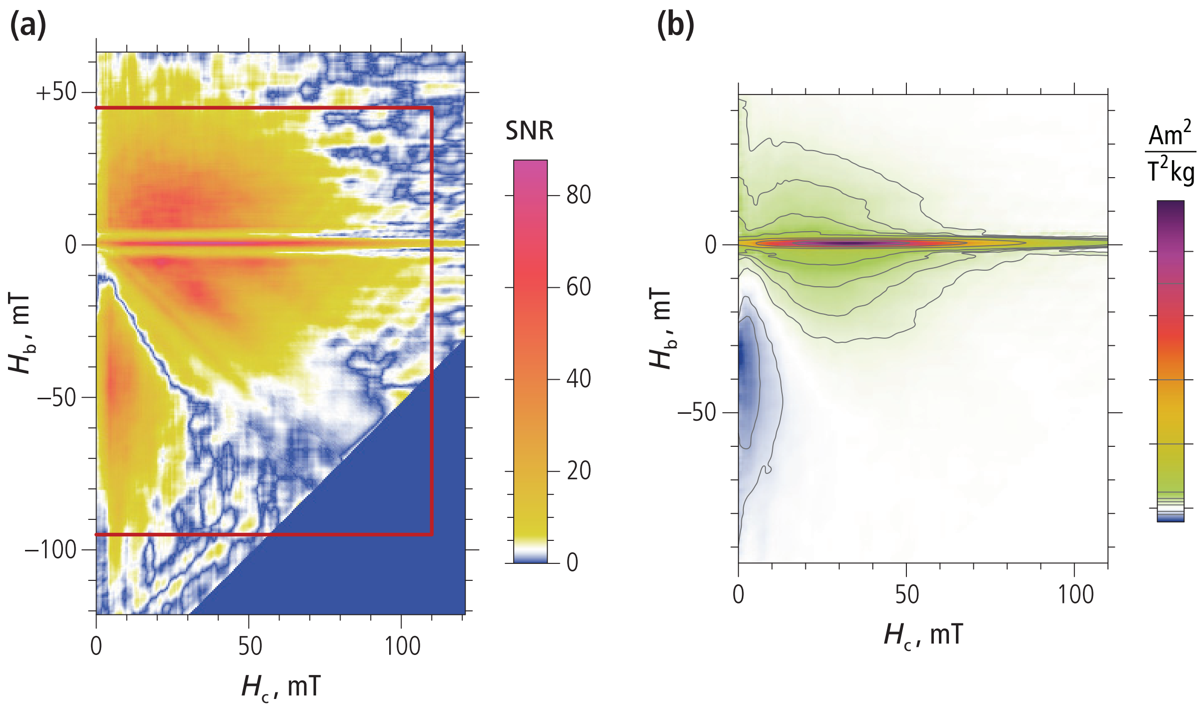


Fig. 5.3: Range selection example based on the FORC diagram of a pelagic carbonate (see the downloadable example "pelagic carbonate"). **(a)** Signal-to-noise-ratio (SNR) generated by `CalculateFORC` during FORC processing. Significant FORC amplitudes are characterized by SNR > 3 (i.e. yellow, red, and purple pixels) and do not cover the whole measurement range. **(b)** The FORC diagram plot has been limited to the red rectangle highlighted in (a) by setting the H_c-range to 0-110 mT (`INPUT 01`), and the H_b-range to −95-45 mT (`INPUT 02`).

In order to facilitate custom FORC space specifications, `PlotFORC` prints a summary table of the original FORC space covered by the FORC matrix, along with the plotted space, e.g.:

```
Summary of FORC space limits:
Original Hc-range: from 0    to 121 (242 points)
Output Hc-range  : from 0    to 110 (220 points)
Original Hb-range: from -121 to 63  (369 points)
Output Hb-range  : from -95  to 45  (279 points)
```

- It is generally recommended to calculate FORC diagrams over the whole range covered by measurements and limit the final range with `PlotFORC`. In this way, diagnostic plots generated by `CalculateFORC` (e.g. the signal-to-noise ratio) can be used for selecting significant FORC regions to be plotted.
- Some parts of the FORC diagram might be confused with a zero background while containing small but significant FORC amplitudes. Therefore, always check the significance of FORC amplitudes with signal-to-noise ratio or standard error plots generated by `CalculateFORC` and other VARIFORC modules.
- `PlotFORC` offers options for customizing the color scale so that significant FORC amplitudes are always distinguishable from zero (see `INPUT 07`).

INPUT 03. Diagonal FORC space limits

The maximum extension of FORC functions is delimited by a triangle with vertices $(\pm H_s, 0)$ and $(0, H_s)$, where H_s is the saturation field, i.e., the positive field above which the major hysteresis loop becomes closed (see Chapter 8). Current FORC measurement protocols reproduce this limit only over the lower quadrant, so that the FORC function might be intrinsically zero over the upper right corner of the measured FORC space (e.g. Fig. 5.4a). In general, if the H_c-axis of a FORC diagrams covers the entire coercivity range, the upper and lower right corners do not display real signals. These regions can be set exactly to zero using diagonal limits of the FORC space. This option is useful for excluding the unnecessary representation of artifacts eventually occurring near the edges of the measurement range.

Diagonal FORC space limits are controlled by `INPUT 03`. Diagonal limits are avoided by setting

```
INPUT 03. Diagonal FORC space limits ....; None
```

In this case, the whole FORC matrix is plotted over the rectangular range chosen through `INPUT 01` and `INPUT 02` (e.g. Fig. 5.3).

Diagonal limits are specified by a pair of numbers referring to the intercepts of the ascending and descending diagonal with the H_c-axis (i.e. at $H_b = 0$), respectively. In most cases, the two limits are equal and coincide with H_s, e.g.

```
INPUT 03. Diagonal FORC space limits ....; 120, 120
```

(Fig. 5.4b), so that FORC amplitudes are set to zero over the saturation region. In the example of Fig. 5.4b, the 120 mT intercepts of the two diagonal limits with the H_c-axis lie outside the diagram.

Different diagonal limits for the upper and lower quadrant can also be chosen. For example,

```
INPUT 03. Diagonal FORC space limits ....; 120, None
```

can be used to exclude possible FORC processing artifacts near the lower diagonal limit of FORC measurements. In this case, the contribution of FORCs beginning at reversal fields < -120 mT is set to zero.

- In some cases, processing artifacts appear along the lower diagonal limit of FORC measurements. You can exclude this region right away during FORC processing with `Calculate FORC` (Chapter 4) or use `PlotFORC` with diagonal limits set by `INPUT 03`.
- Use the diagnostic plots generated by `CalculateFORC` to establish a significant FORC range to be plotted.

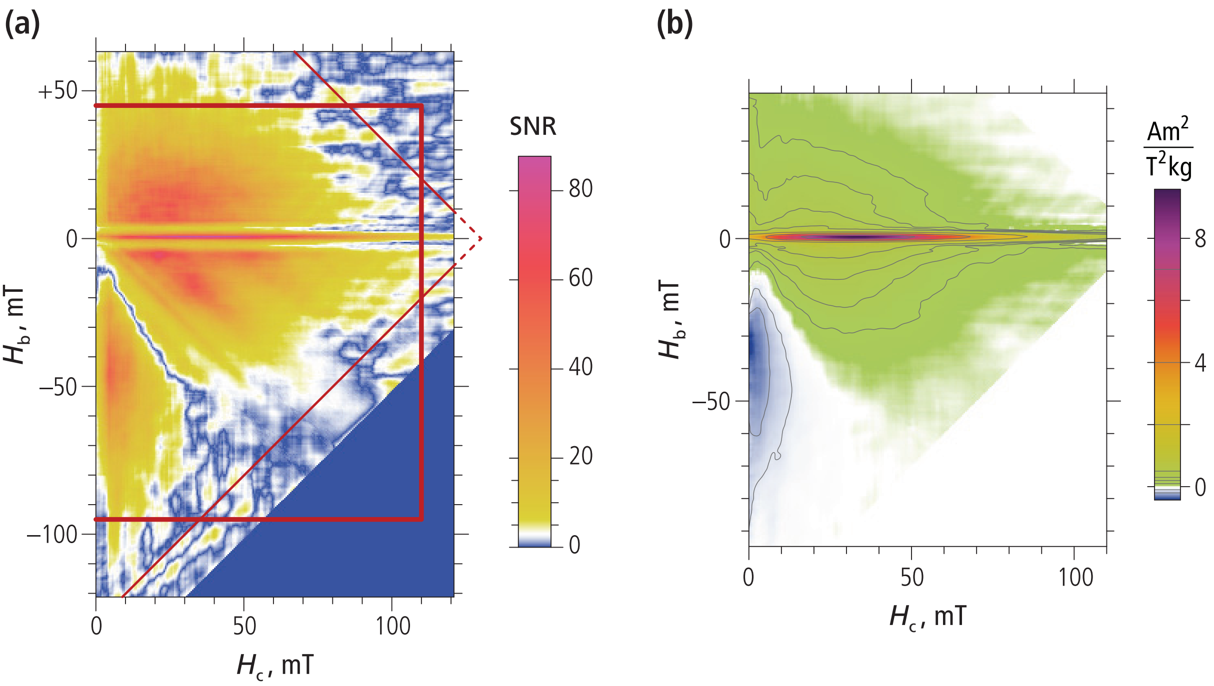


Fig. 5.4: Example of plotted FORC range limitation along diagonals, based on FORC data from a pelagic carbonate (see the downloadable example "pelagic carbonate"). **(a)** Signal-to-noise-ratio (SNR) generated by `CalculateFORC` during FORC processing. Significant FORC amplitudes are characterized by SNR > 3 (i.e. yellow, red, and purple pixels). **(b)** The plotted FORC diagram has been limited to the red rectangle highlighted in (a) by setting the H_c -range to 0-110 mT (`INPUT 01`), and the H_b -range to −95-45 mT (`INPUT 02`). Furthermore, diagonal limits intercepting the H_c -axis at 120 mT have been added (thin red lines). The FORC function is set to zero outside of these limits, regardless of the actual FORC matrix values. In this diagram, small FORC amplitudes have been strongly enhanced in order to highlight the diagonal limits.

INPUT 04. FORC unit factor

The VARIFORC unit management system automatically chooses the most suited unit of the FORC function, on the basis of original measurement units entered with `ImportFORC`. Information about the units is contained in the header of the FORC matrix file imported by `PlotFORC`, e.g.

```
Field unit, magnetization unit, data matrix unit
T,mAm2/kg,1e-3 Am2/(T2 kg)
```

in case of the FORC diagram shown in Fig. 5.5. The field and magnetization units taken by `PlotFORC` in this example are therefore T and mAm^2/kg, respectively, and the corresponding unit of the FORC matrix is (magnetization)/(field)^2, i.e. $mAm^2/(T^2kg)$.

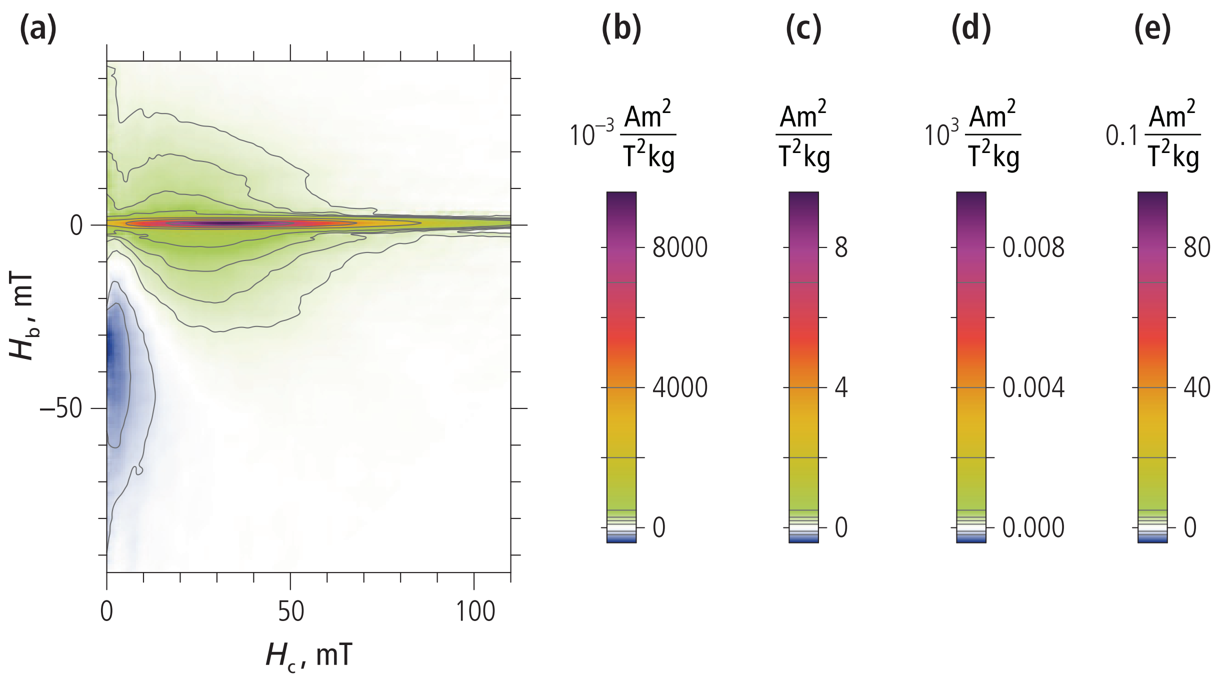


Fig. 5.5: (a) FORC diagram of a pelagic carbonate (see the downloadable example "pelagic carbonate"). **(b-e)** Color scales for the FORC diagram in (a), with units obtained from different settings of `INPUT 04`, i.e.: `1` (b), `Automatic` or `1e3` (c), `1e6` (d), and `100` (e). The automatic option (c) chooses the SI multiplier that yields a scale with single-digit FORC values.

FORC amplitudes expressed with original units can be very small or very large. In order to avoid bulky numbers in the color bar labeling of the plotted FORC diagram, the original FORC unit from the FORC matrix file can be multiplied by a power-of-ten as specified with `INPUT 04`. The recommended option is

```
INPUT 04. FORC unit factor ....; Automatic
```

In this case, `PlotFORC` chooses the power-of-ten multiplier that gives color scale labels with the least number of digits. In the example of Fig. 5.5c, this multiplier is 10^3, so that the unit of the plotted

FORC diagram is 10^3 mAm^2/(T^{2}kg), i.e. simply Am^2/(T^{2}kg).

Alternatively, any desired unit multiplier can be specified with `INPUT 04`, e.g.

```
INPUT 04. FORC unit factor ....; 1e6
```

for transforming the original FORC unit mAm^2/(T^{2}kg) into kAm^2/(T^{2}kg) (Fig. 5.5d).

- `INPUT 04` affects only the plotted FORC unit. Plotting parameters, such as contour level specifications (`INPUT 11`), must be expressed in original units, as specified in the FORC matrix file.
- Summary tables produced by `PlotFORC` (e.g. the range of FORC values) are expressed in original units. Plotting parameters such as contour levels can be chosen on the basis of such tables.

- Explicit unit multiplies can be entered to match the plotted unit with that of other FORC diagrams.
- If comparisons with other FORC diagrams are not needed, `INPUT 04` is best set to `Automatic`.

INPUT 05. Color scale range

`PlotFORC` offers virtually infinite possibilities for controlling the color scale of plotted FORC diagrams through options entered with `INPUT 05`, `INPUT 06`, and `INPUT 07`. Optimized color scales are essential for representing small but significant FORC magnitudes so that they are not confused with the noise background. The first parameter, `INPUT 05`, controls the range of FORC values covered by the chosen color scale. The simplest option for `INPUT 05` is

```
INPUT 05. Color scale range .....; All
```

in which case the color scale extends exactly from the minimum to the maximum value of the FORC function (Fig. 5.6b). Alternatively, arbitrary color scale ranges can be entered with two numbers corresponding to minimum and maximum FORC values. For example,

```
INPUT 05. Color scale range .....; -200, 1000
```

defines a color scale ranging from −200 to 1000 units, regardless of the range effectively covered by the FORC function. Numerical values in `INPUT 05` are tied to the original FORC units of the FORC matrix file, regardless of the unit chosen for plotting the FORC diagram (see `INPUT 04`).

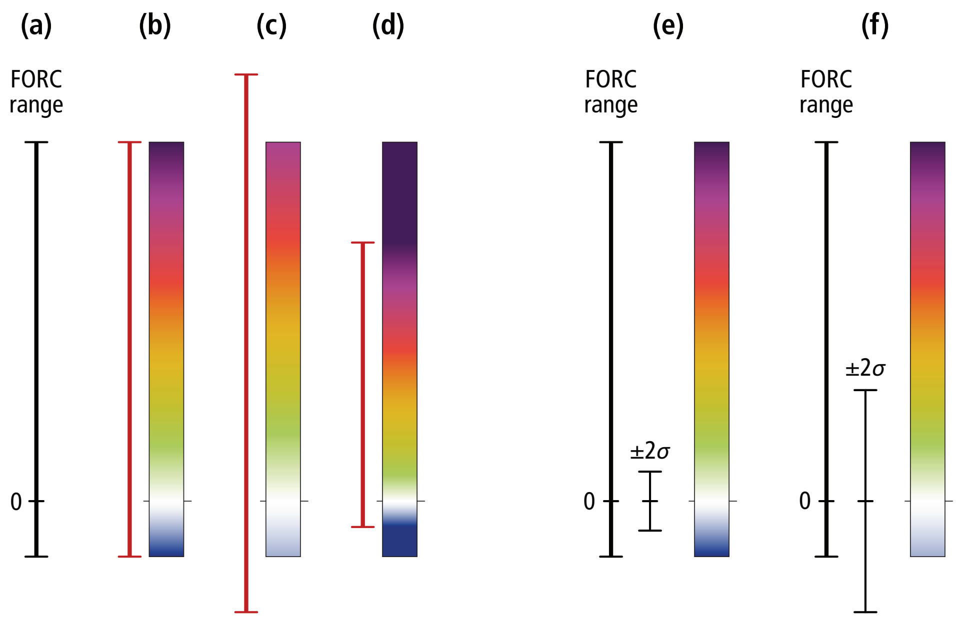


Fig. 5.6: (a) Range of FORC values, represented by a vertical bar. **(b-d)** Color scale examples with color ranges (red vertical bars) that are coincident with (b), exceeding (c), and included in (d) the FORC range. If the color range does not cover all FORC values, as in (d), the exceeding FORC range is represented with the end colors of the scale. With the color scale chosen for this figure ("Iridescent", see `INPUT 06`), zero is coded in white, independently of the chosen color range. **(e-f)** Automatic color ranges (i.e. `INPUT 05` set to `Automatic`) for the case of (e) a FORC diagram where all values are significant, and (f) a FORC diagram where negative values are not significant. The vertical bar labeled with $\pm 2\sigma$ indicates the double standard error of minimum (lower half) and maximum (upper half) FORC values. The color scale in (f) begins with -2σ, instead of the minimum value of the FORC function.

If the color scale range specified with `INPUT 05` exceeds the minimum or maximum FORC values, FORC diagrams will be represented with a subset of the color scale (Fig. 5.6c). If, on the other hand, the range specified with `INPUT 05` does not cover all FORC values, parts of the FORC diagram exceeding the given color range will be represented with minimum/maximum color scale values (Fig. 5.6d). There are several reasons for entering a color scale range that does not coincide with the FORC range. For example, a wider color scale range can be chosen in order to avoid colors corresponding to the beginning and/or the end of the chosen color scale (Fig. 5.6c). On the other hand, a reduced color scale range can be chosen in order to enhance small FORC amplitudes (Fig. 5.6d).

The color range can be also entered as percent values of the maximum FORC value, just by adding the % symbol at the end, e.g.:

```
INPUT 05. Color scale range .....; -10,90%
```

for a color scale ranging from −10% to 90% of the maximum FORC value. This format, which is available since version 2.03, is particularly useful for comparing FORC diagrams through homogeneous plotting criteria. For example, the same level of color saturation for negative values can be reached by setting the first limit of the color range to a fixed percent value, e.g. −10%.

Finally, the color range can be chosen automatically by setting

```
INPUT 05. Color scale range .....; Automatic
```

This option usually yields best results and is highly recommended, especially in combination with color scales where zero values are coded in white, such as the one shown in Fig. 5.6. With this option, the color scale covers the whole range of FORC values, as well as the double standard errors, $\pm 2\sigma$, of minimum and maximum values. If standard errors are small (i.e. 2σ does not exceed the corresponding FORC amplitudes), the color scale simply coincides with the FORC range (Fig. 5.6e). If, on the other hand, $\pm 2\sigma$ exceeds minimum or maximum FORC values, the color scale covers the double standard error (Fig. 5.6f). This choice ensures that small, non-significant FORC amplitudes (usually negative ones) are only barely distinguishable from zero.

The use of color scale ranges is quite intuitive, as illustrated with the example of Fig. 5.7. In this example, the FORC diagram has been limited for illustration purposes to a region where negative values are not significant. If `INPUT 05` is set to `All`, negative FORC values, which do not add any information to the diagram, are unnecessarily highlighted (blue spots in Fig. 5.7a). This problem is eliminated by setting `INPUT 05` to `Automatic` (Fig. 5.7b), so that `PlotFORC` extends the color range enough for the blue end to coincide with the double standard error. Because negative amplitudes are much smaller than corresponding standard errors, they are assimilated with a zero background (i.e., white). Finally, an explicit specification of the color scale range can be used for covering only weak FORC amplitudes (Fig. 5.7c): in this case, the central ridge is completely saturated. A better solution is available for this purpose (see `INPUT 07`).

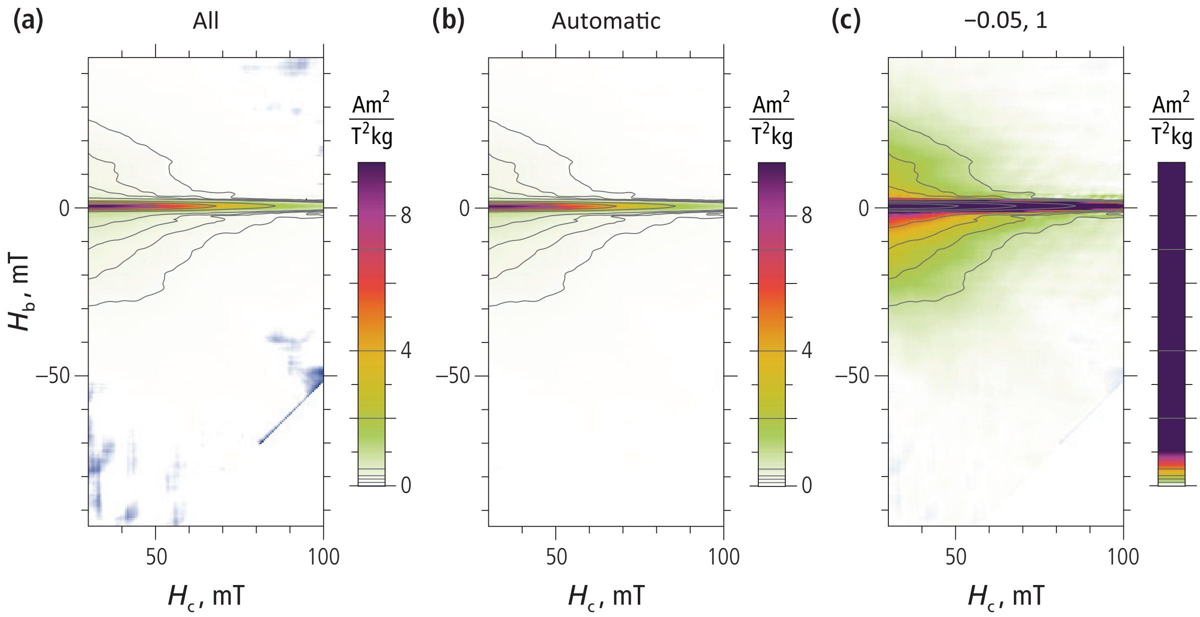


Fig. 5.7: Color scale ranging examples based on the FORC diagram of a pelagic carbonate (see the downloadable example "pelagic carbonate"). This is the same diagram of Fig. 5.5, except for the fact that a smaller FORC range has been chosen in order to exclude significant negative amplitudes for demonstration purposes. The range of plotted FORC values is comprised between −0.013±0.3 and +9.58±0.21 $Am^2/(T^2kg)$. All plotting parameters are the same for the three examples, except for `INPUT 05`, which was set to **(a)** `All`, so that the color range coincides with the plotted FORC range, **(b)** `Automatic`, so that the color range is extended to include the double standard error of minimum and maximum FORC values, and **(c)** `−50,1000` (in $mAm^2/(T^2kg)$, corresponding to −0.05, 1 on the color scale) so that small positive amplitudes <1 $Am^2/(T^2kg)$ are highlighted.

In order to facilitate explicit color range specifications, `PlotFORC` prints a summary table of the FORC range and corresponding double standard errors (if available), e.g.:

```
Range parameters for color scale Iridescent in original units:
FORC range .........: from -429.391 ±100.143 to 9580.5 ±210.068
Color scale range ..: from -429.391          to 9580.5
Green at ...........: 429.391
```

where the first row gives the range of plotted FORC values and the remaining rows shows the parameters controlling the chosen color scale. In the above example, negative FORC amplitudes are clearly not significant and the lower limit of the color scale range coincides with the double standard error.

- Some `PlotFORC` options (e.g. `Automatic` for `INPUT 05`) are based on FORC error estimates stored in a separated file located in the same folder as the FORC matrix and ending with `_FORCStandardError_VARIFORC.txt`. `PlotFORC` automatically imports the error file corresponding to the chosen FORC data. Therefore, never delete or move error files.
- If `PlotFORC` does not find the error file corresponding to the chosen FORC matrix, it will disable or modify all options that depend on error estimates. For example, automatic color scale ranging is disabled.

- It is always recommended to perform a first `PlotFORC` run with automatic color scale ranging (i.e., `INPUT 05` set to `Automatic`). In general, FORC diagrams generated with the automatic option do not require further adjustments.
- Explicit color range specifications (i.e., `INPUT 05` set to a number pair) can be used to plot several FORC diagrams with exactly the same color scale for comparison purposes. In this case, all diagrams should be processed with identical `INPUT 05` settings.

INPUT 06. Color function specification

A typical characteristic of many FORC diagram is the coexistence of small regions with very large, positive amplitudes (e.g. ridges, peaks) and large areas with very small positive or negative amplitudes. Standard color scales, such as the commonly used MATLAB® "Jet", perform very poorly, because details over the low-amplitude regions get completely lost (Fig. 5.8a). Furthermore, the presence of negative amplitudes in some diagrams shifts the origin of the color scale, giving very different appearances to similar diagrams. Finally, many color scales contain colors lying outside the printable gamut, yielding poor printing results. For these reasons, printable color functions with high rendering capability have been specially developed for use with `PlotFORC` (e.g. Fig. 5.8b).

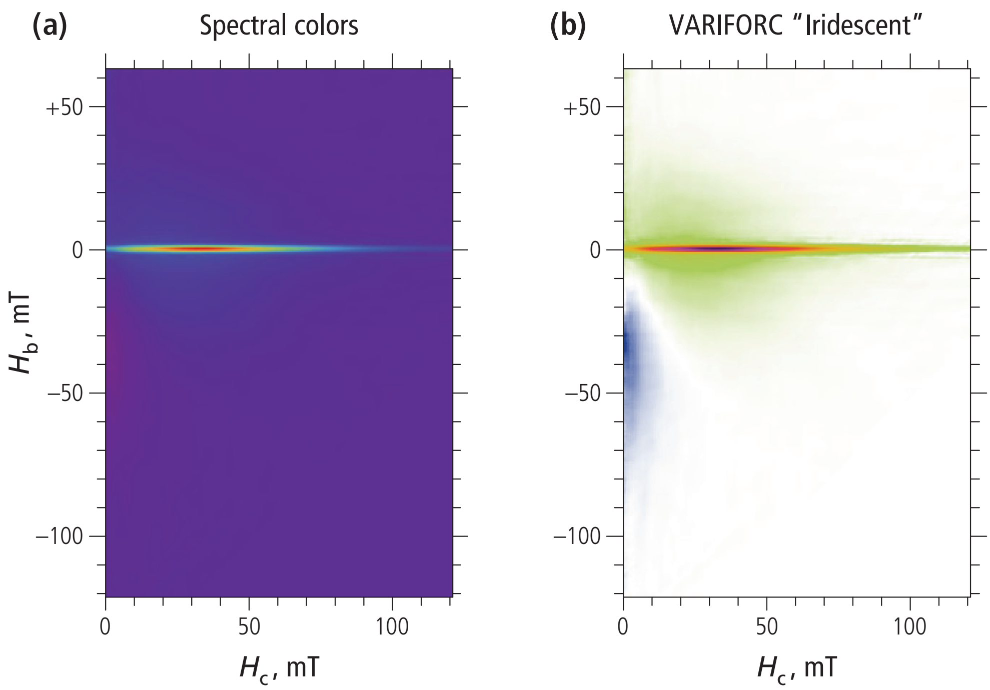


Fig. 5.8: Color scale examples based on the FORC diagram of a pelagic carbonate sample (see the downloadable example "pelagic carbonate"). **(a)** Representation with a common standard color scale based on spectral colors (e.g. MATLAB® "Jet" and Mathematica® "Rainbow"). **(b)** `PlotFORC` representation with the color scale "Iridescent" and default color scale settings. Low-amplitude FORC features (e.g. negative amplitudes in the lower quadrant and high-coercivity tail of the central ridge) are barely recognizable in (a), but clearly visible in (b).

`PlotFORC` has four built-in color functions: two of them, named "Iridescent" and "Sunsky", are optimized for color print and monitor representations, while the other two, named "Aquamarine" and "Sepia", are monochrome scales best suited for black-and-white printing. All color scales are based on blending of Pantone® process colors and are therefore suited for printing. Color functions and optional control parameters are entered with `INPUT 06`. The recommended choice is

```
INPUT 06. Color function specifications ....; Iridescent
```

which has been used to produce the example of (Fig. 5.8b). Other options for `INPUT 06` are `Sunsky` (color), `Aquamarine` (monochrome), and `Sepia` (monochrome). FORC diagram examples produced with these color scales are shown in Fig. 5.9.

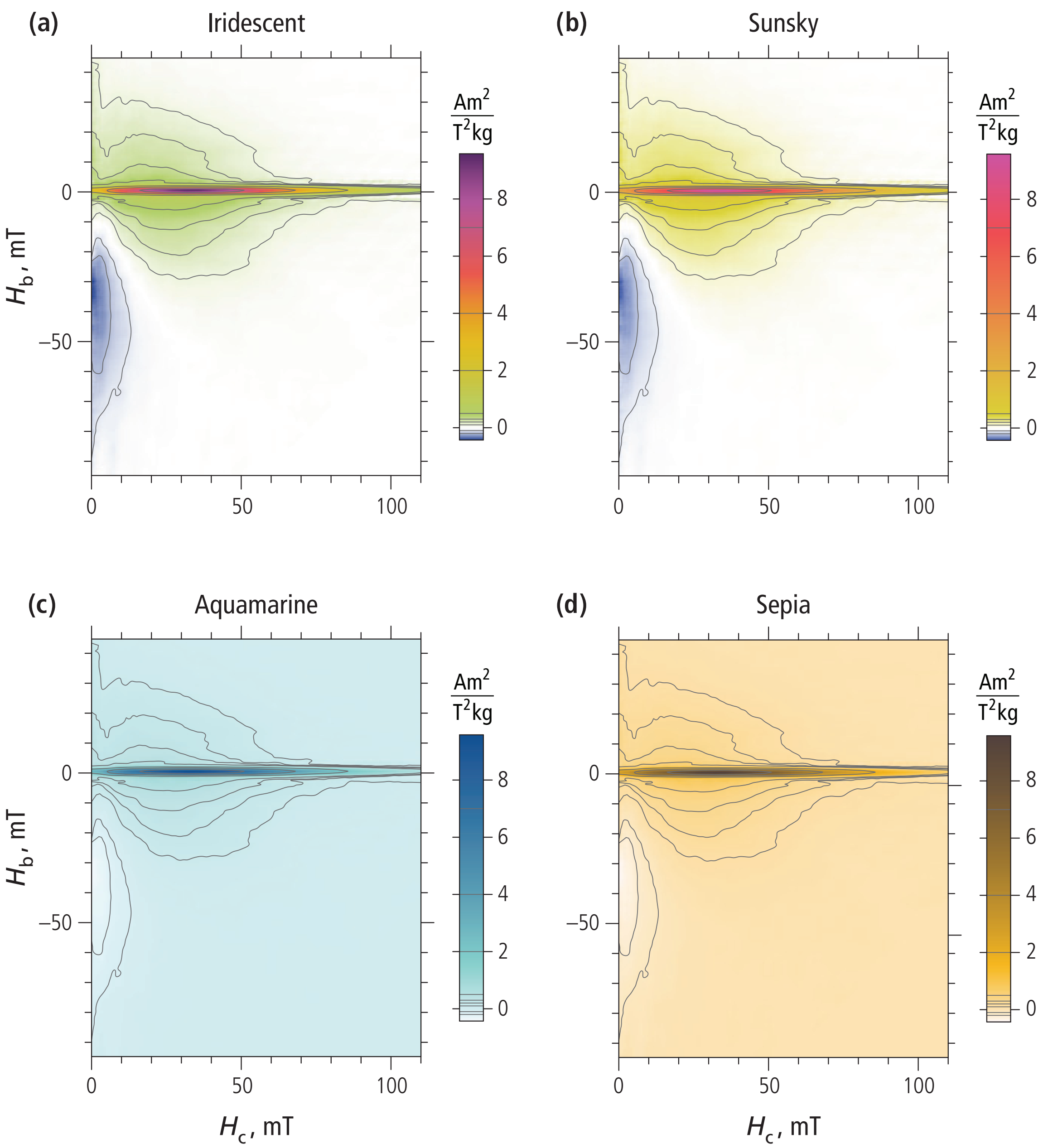


Fig. 5.9: `PlotFORC` color scales, shown for the same FORC diagram of a pelagic carbonate sample (see the downloadable example “pelagic carbonate”). The color scale was chosen by setting `INPUT 06` to the corresponding color scale name (shown at the top of each diagram) with a color saturation of 0.9. “Iridescent” has the highest rendering capability.

`Iridescent` has the highest rendering capability and is the color scale that provides the least ambiguous conversion to gray tones after monochrome scales. It is therefore the recommended default choice for presentations and professional printing.

Each color function has default settings that perform satisfactory in most cases. Different settings can be optionally specified with `INPUT 06`. The first option is color saturation, which is expressed by a number comprised between 0 (no saturation: the whole color scale is white) and 1 (maximum saturation of the Pantone® reference colors on which the color scale is built). The default choice is 0.9. Saturation specifications can be entered after the color function name, separated by a comma, e.g.

```
INPUT 06. Color function specifications ....; Iridescent, 0.9
```

for using the color function “Iridescent” with 90% color saturation (e.g. Fig. 5.9a). The default 90% saturation is recommended for preparing article figures, since it fits the gamut of most color printers (Fig. 5.10), while 100% color saturation best suited for monitors and high-quality printing.

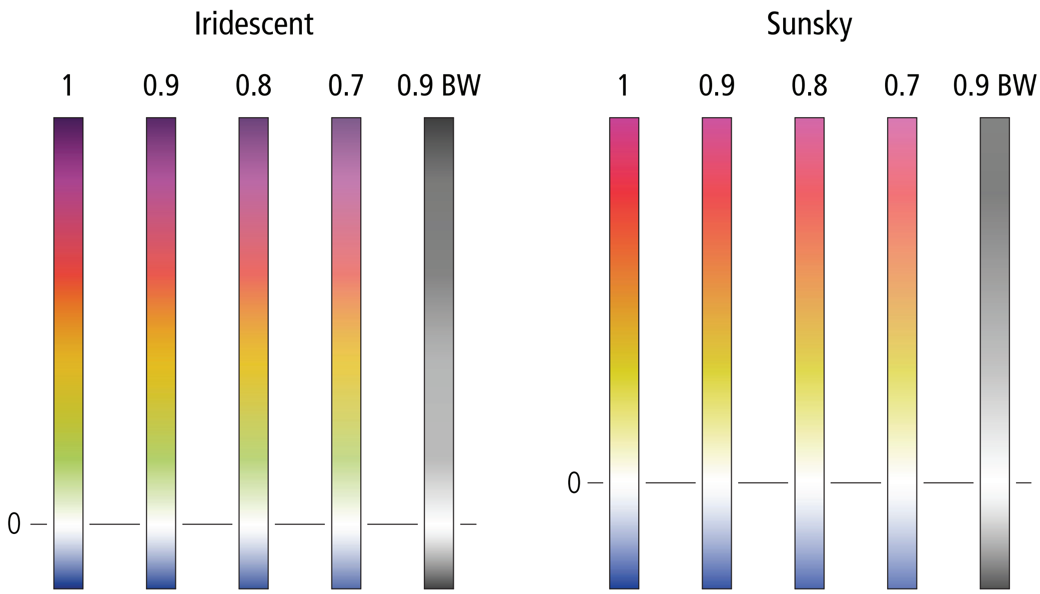


Fig. 5.10: `Iridescent` and `Sunsky` color scales with color saturation values of 1, 0.9, 0.8, and 0.7, respectively. The scale resulting from conversion of 90% saturation to black-and-white printing (0.9 BW) is shown for comparison. The color scales have a fixed zero point corresponding to pure white (horizontal line). `Iridescent` ensures monotonic transitions from white to dark gray for both positive and negative amplitudes, and is therefore suited to black-and-white printing of FORC diagrams where negative and positive amplitudes can be identified unambiguously.

Monochrome color functions (i.e. `Aquamarine` and `Sepia`) with saturations of the order of 70% are best suited for black-and-white printing if the ambiguity of `Iridescent` colors after grayscale conversion is unacceptable (Fig. 5.11). Both monochrome functions yield identical gray tones.

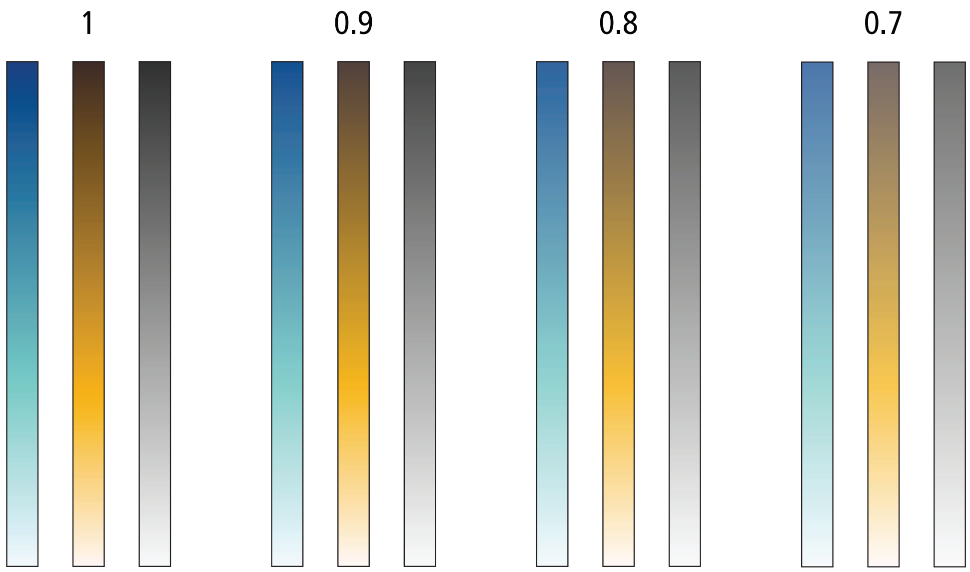


Fig. 5.11: Monochrome color scales `Aquamarine` and `Sepia`, and their grayscale conversions, with color saturation levels given by the top numbers. Both scales convert to the same gray tones and give same black-and-white printing results. Unlike `Iridescent` and `Sunsky`, these scales do not have a fixed color for zero values.

A further setting for all color scales is their so-called gamma correction, which is optionally entered with `INPUT 06` as second numeric parameter after color function name and color saturation. For example,

```
INPUT 06. Color function specifications ....; Iridescent, 0.9, 1
```

gives the default gamma correction of 1 and is equivalent to

```
INPUT 06. Color function specifications ....; Iridescent, 0.9
```

Gamma correction is defined as the positive exponent of a power law governing the transition between white and the next reference color (i.e. blue and green for `Iridescent`, blue and yellow for `Sunsky`, greenish blue for `Aquamarine`, and brown for `Sepia`). Gamma corrections >1 expand light tones close to white, while gamma corrections <1 produce the opposite effect (Fig. 5.12). The gamma correction is used for regulating color transitions produced by the monochrome color functions `Aquamarine` and `Sepia`; its use with `Iridescent` and `Sunsky` is possible but not recommended.

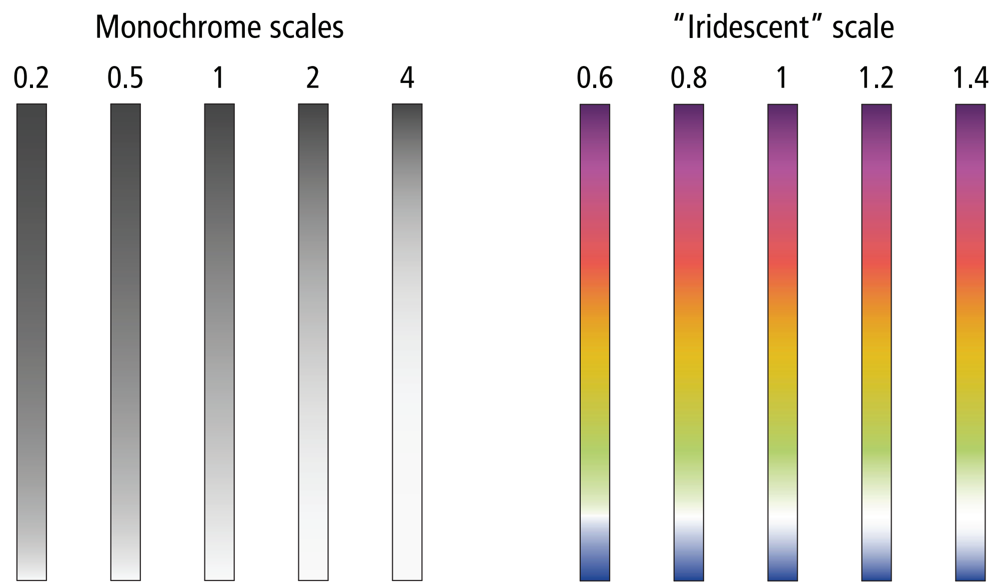


Fig. 5.12: Gamma correction of monochrome color functions after conversion to gray tones (`Aquamarine` and `Sepia`) and of the `Iridescent` color function, with gamma exponents given by numbers on the top of each scale. A gamma exponent of 1 gives a perceptually homogeneous transition from white to dark gray. All scales have been produced with 90% color saturation.

The default gamma exponent of 1 produces a perceptually homogeneous transition from white to the dark end of monochrome color scales (i.e., `Aquamarine` and `Sepia`). This is of little importance once the color function is applied to the FORC diagram, because the abundance of light and dark colors depends on balance between large and small amplitudes of the FORC function. In some cases, the default gamma exponent of monochrome color scales gives good representations of the FORC function, especially if special features such as the central ridge are absent. In other cases, large areas of the FORC diagram corresponding to low amplitudes are represented with insufficient contrast (Fig. 5.13a), which is enhanced by choosing gamma exponents <1 (Fig. 5.13b).

- Never use the gamma correction option for enhancing the rendering of small FORC amplitudes when using non-monochrome color functions, i.e. `Iridescent` and `Sunsky`. The rendering enhancement of these color functions is performed automatically by `PlotFORC` and is controlled by `INPUT 07`.
- The rendering capability of monochrome color functions, i.e. `Aquamarine` and `Sepia`, is intrinsically low and is only marginally improved by gamma corrections. When possible, use the `Iridescent` color function also for black-and-white printing. An example is shown in Fig. 5.14.

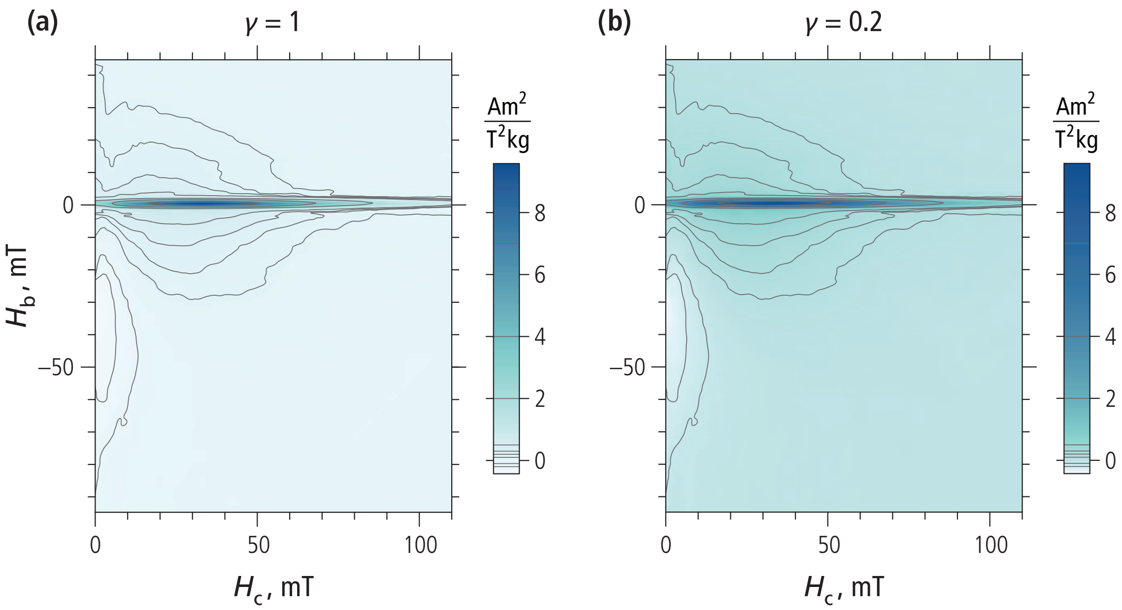


Fig. 5.13: Gamma correction of the monochrome color scale `Aquamarine`, applied to the FORC diagram of a pelagic carbonate sample (see the downloadable example "pelagic carbonate"). **(a)** The default gamma value of 1 provides almost no contrast outside the central ridge. Negative amplitudes are barely recognizable. **(b)** The contrast of small FORC amplitudes is enhanced by choosing a gamma correction <1, in this example, 0.2. Negative values are now recognizable, but the performance of this color function remains poor in comparison with the color scales `Iridescent` and `Sunsky`.

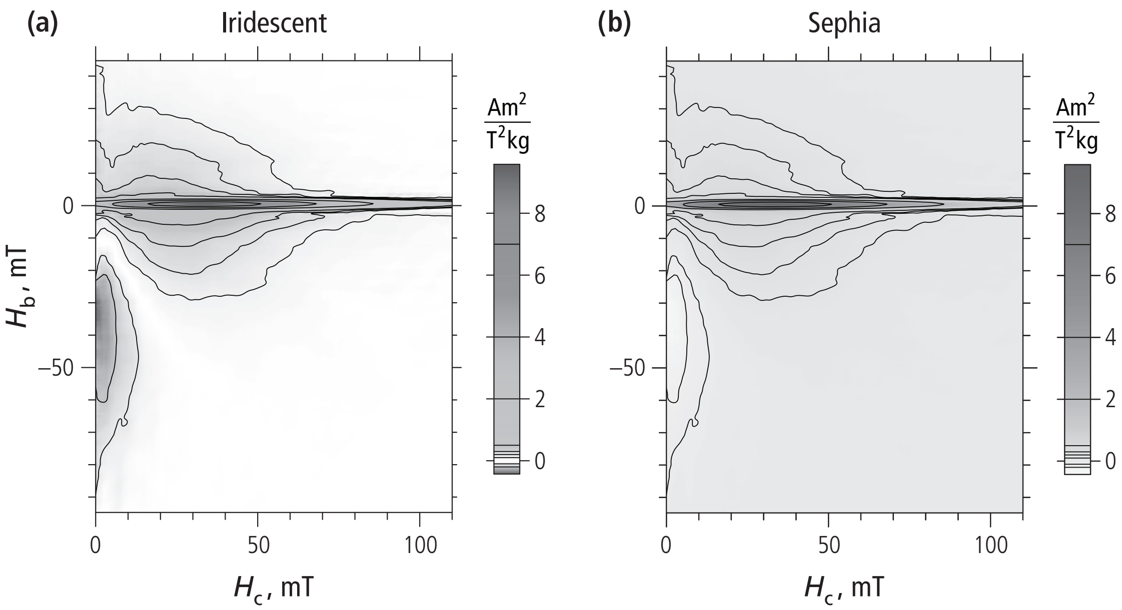


Fig. 5.14: Grayscale conversion of FORC diagrams of a pelagic carbonate sample (see the downloadable example "pelagic carbonate") produced with **(a)** the color function `Iridescent` at 90% color saturation and default settings, and **(b)** the color function `Sepia` at 90% color saturation and a gamma correction exponent

of 0.5. `Iridescent` maintains a higher rendering performance even in black-and-white printing, provided that positive and negative FORC amplitudes are clearly distinguishable, as in (a).

The color function `Iridescent` maintains its high rendering performance even in black-and-white printing (Fig. 5.14). The only drawback with respect to monochrome color scales is related to the fact that negative and positive FORC amplitudes are represented with same gray tones and can be confused. Therefore, `Iridescent` should be used for black-and-white printing only when positive and negative regions of the FORC diagram are clearly distinguishable, as in Fig. 5.14a.

- The simplest color scale options are selected by just entering the color scale name with `INPUT 06`. In this case, color saturation and gamma correction exponent are both set to 1.
- An explicit color saturation specification is optionally entered after the color function name: for example, `Sepia,0.8` means that the saturation of the “Sepia” color scale is set to 80%.
- An explicit gamma correction specification is optionally entered after saturation: for example, `Sepia,0.9,0.5` means that a gamma correction exponent of 0.5 is applied to the 90%-saturated “Sepia” color scale. Color saturation must always be entered before the gamma correction specification.
- Best results for presentations and printing are obtained with `Iridescent` and no further specifications (i.e. color saturation is 90% and the gamma correction exponent is 1). This `INPUT 06` specification can be used as default color scale for most applications.
- Small gamma corrections to `Iridescent` and `Sunsky` color scales are only useful for fine-tuning of printing results.
- A color saturation of 0 can be used in combination with contour drawing (`INPUT 11`), in order to obtain a contour diagram without shading. In this case, the color scale is white.

INPUT 07. Expanded color range

INPUT 07 provides additional options for the non-monochrome color functions Iridescent and Sunsky and has no influence on monochrome color functions. The recommended default option is

```
INPUT 07. Expanded color range option ....; Automatic
```

and provides optimal results in almost every case. Different options can be entered for fine-tuning purposes, and to fix a color function when different FORC diagrams should share the exact same color scale.

INPUT 07 controls the extension of a sub-range of the whole color scale that is designed to cover low FORC amplitudes: this range is called *expanded color range*. The expanded color ranges of Iridescent and Sunsky cover the blue-white-green and blue-white-yellow color transitions, respectively (Fig. 5.15), where white is assigned to zero values.

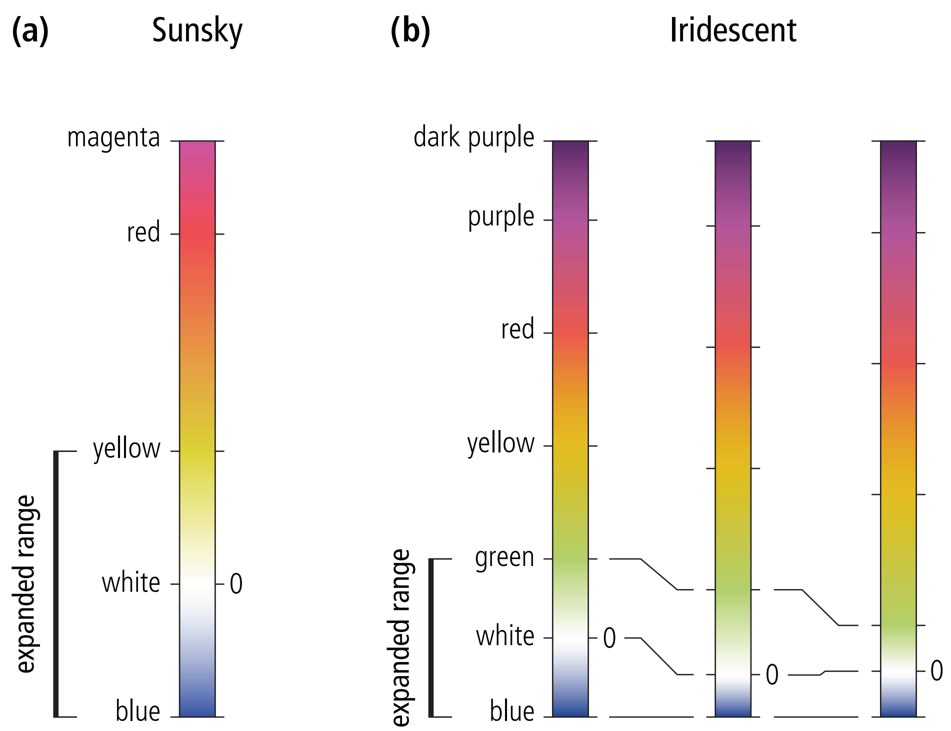


Fig. 5.15: Expanded color ranges of the PlotFORC color scales Sunsky and Iridescent. **(a)** The expanded color range runs from blue to yellow through white. **(b)** The expanded color range runs from blue to green through white. The left scale in (b) corresponds to the unmodified color function Iridescent. The two color scales to the right were obtained by fitting the expanded color range to a given range of positive and negative FORC amplitudes.

The `Iridescent` and `Sunsky` color functions are defined on the basis of optically homogeneous transitions between reference colors. The default position of reference colors on the scale can be used without further modification by setting `INPUT 07` to `Fixed`. This option is not recommended, because the color function is fixed between minimum and maximum values, regardless of the distribution of FORC values in between. Consequently, small FORC amplitudes are often plotted with insufficient color contrasts (Fig. 5.16a). Best results are obtained with

```
INPUT 07. Expanded color range option ....; Automatic
```

in which case the color function is automatically adapted to the characteristics of the FORC function by manipulating the expanded color range (Fig. 5.16b). For example, the negative part of the expanded color range (i.e., all colors between blue and white) is scaled to match the range of negative FORC amplitudes, according to the color range specifications given with `INPUT 05`. Adjustment of the positive half of the expanded color range (i.e., all `Iridescent` colors between white and green) is based on a similar principle.

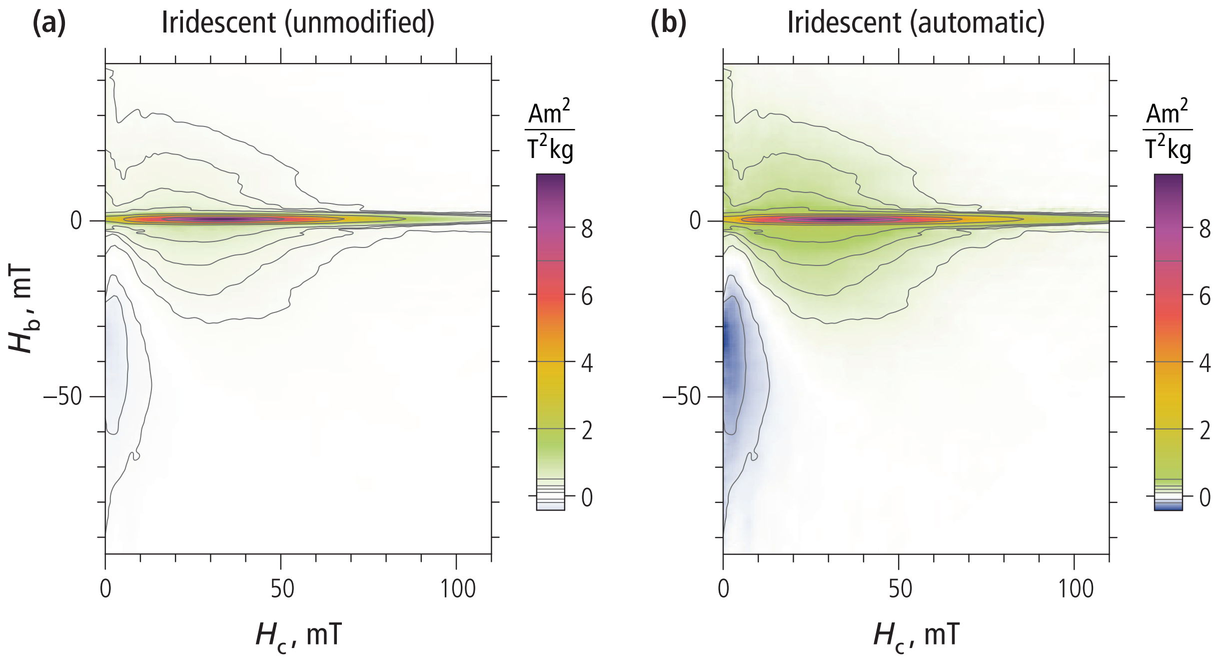


Fig. 5.16: `Iridescent` color scale examples based on the FORC diagram of a pelagic carbonate sample (see the downloadable example "pelagic carbonate"). **(a)** The color scale is used without further adjustments by setting `INPUT 07` to `None`. In this case, white and the purple end of the color scale coincide with zero and maximum FORC values, respectively. Small FORC amplitudes are plotted with insufficient color saturation. **(b)** Adjusted `Iridescent` color scale obtained by setting `INPUT 07` to `Automatic`. In this case, both ends of the color scale coincide with the FORC range, while zero amplitudes are still shown in white, as in (a). Notice the color saturation increase for negative amplitudes, and the small positive contributions around the central ridge.

In general, if INPUT 07 is set to Automatic, PlotFORC tries to make the expanded color range symmetric about zero (Fig. 5.17b). The rationale for a symmetric extended range is tied to negative FORC amplitudes, which, in case of single-domain particles, reflect magnetic moment rotations in the applied field. Such rotations produce antisymmetric positive and negative amplitudes over the lower quadrant of the FORC diagram [*Newell*, 2005], which are ideally covered by the expanded color range.

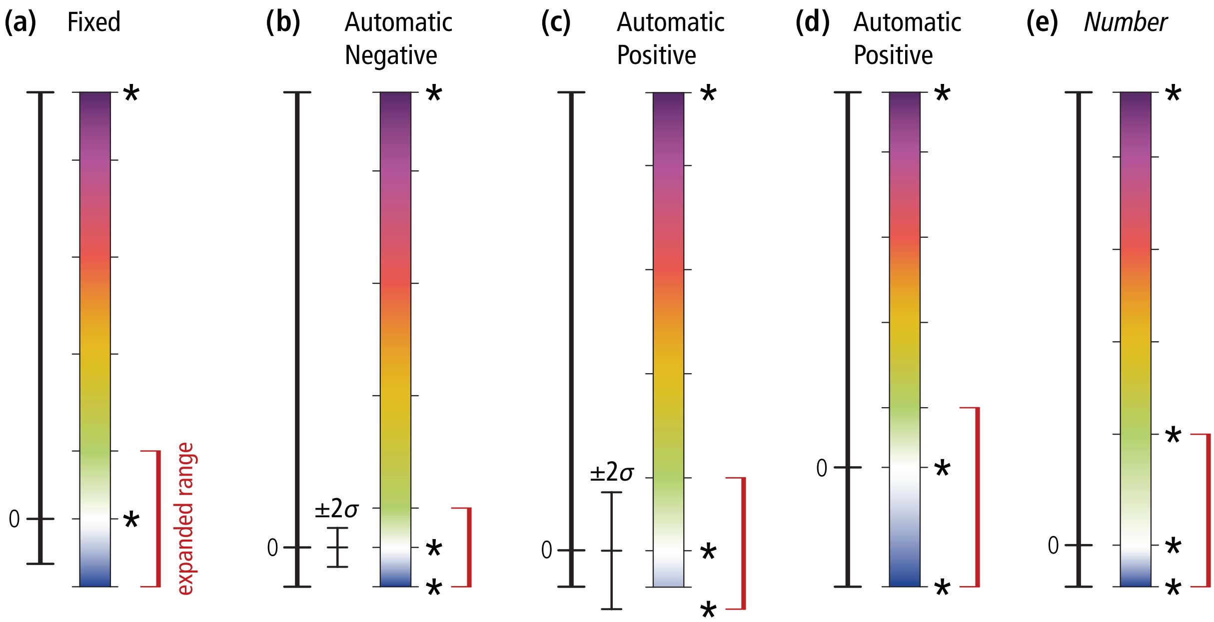


Fig. 5.17: Adjustment of the color scale Iridescent for best match with FORC diagram properties. In all five examples, the vertical bar on the left represents the range of FORC amplitudes, followed by the color scale obtained with the INPUT 07 option indicated on the top (i.e. Fixed, Automatic, Negative, and Positive). The expanded color range (blue-white-green) is highlighted in red. The asterisks indicate points where the color scale is tied to specific values of the FORC range. **(a)** If INPUT 07 is set to Fixed, the expanded range remains unmodified and the color scale is tied at 0 and at the maximum FORC amplitude specified with INPUT 05. **(b)** If INPUT 07 is set to Automatic and negative FORC amplitudes are significant (i.e. larger than the double standard error $\pm 2\sigma$), the negative end of the expanded color range coincides with the FORC range, and the positive end is chosen so that the expanded range is symmetric about 0. The same result is obtained by setting INPUT 07 to Negative, independently of whether negative FORC amplitudes are significant or not. **(c)** If INPUT 07 is set to Automatic and negative FORC amplitudes are not significant (i.e. smaller than the double standard error $\pm 2\sigma$), the expanded color range remains unmodified. **(d)** If INPUT 07 is set to Automatic and negative FORC amplitudes exceed the unmodified expanded range, the negative part is extended, while the positive part remains unchanged. Setting INPUT 07 to Positive works like the Automatic option for the negative part of the expanded range, while the positive part is always left unchanged. **(e)** The positive limit of the expanded color range can be specified explicitly by entering the corresponding numerical value with INPUT 07.

In some cases, however, maintaining the negative and positive half of the expanded color range symmetric is not meaningful, for example when the FORC function is intrinsically positive, or, on the contrary, when it contains very large negative amplitudes. If negative contributions are absent or produced by noise, `PlotFORC` automatically ignores them and leaves the expanded range of the color scale unmodified (Fig. 5.17c). In case of large negative FORC amplitudes that exceed the unmodified color scale, the negative range is extended, while the positive part of the expanded range is left unmodified (Fig. 5.17d).

The criteria used for setting the positive limit of the expanded color range (i.e., `Iridescent` green and `Sunsky` yellow levels) can be customized. For example,

```
INPUT 07. Expanded color range option ....; Negative
```

makes the expanded color range symmetric, independently of the actual significance or amplitude of negative FORC contributions. On the other hand, the positive part of the color scale is left unchanged by setting

```
INPUT 07. Expanded color range option ....; Positive
```

so that only the negative half of the expanded color range is adapted to the FORC function.

Finally, the positive limit of the expanded color range can be controlled by entering the positive FORC amplitude that should be shown in green (`Iridescent`) or yellow (`Sunsky`), e.g.

```
INPUT 07. Expanded color range option ....; 100
```

(Fig. 5.18b). In this case, be sure that you enter the numerical value in original FORC units, as defined in the FORC matrix file. Explicit numerical specifications of `INPUT 07` can be used to correct the results obtained with the `Automatic` option, for example when representing several FORC diagrams with exactly the same color scale. In this case, `INPUT 07` should be the same for all plots. You can use the FORC range and color range summary produced by `PlotFORC`, e.g.

```
Range parameters for color scale Iridescent in original units:
FORC range .........: from -429.391 ±100.143 to 9580.5 ±210.068
Color scale range ..: from -429.391          to 9580.5
Green at ...........: 429.391
```

as a guide for explicit `INPUT 07` specifications. The last line of the above summary, which was generated with `INPUT 07` set to `Automatic`, gives the positive FORC amplitude that is coded with green (i.e., 429.391 Am^2/T^2). Upon setting this amplitude to `100` with `INPUT 07` the summary becomes:

```
Range parameters for color scale Iridescent in original units:
FORC range .........: from -429.391 ±100.143 to 9580.5 ±210.068
Color scale range ..: from -429.391          to 9580.5
Green at ...........: 100
```

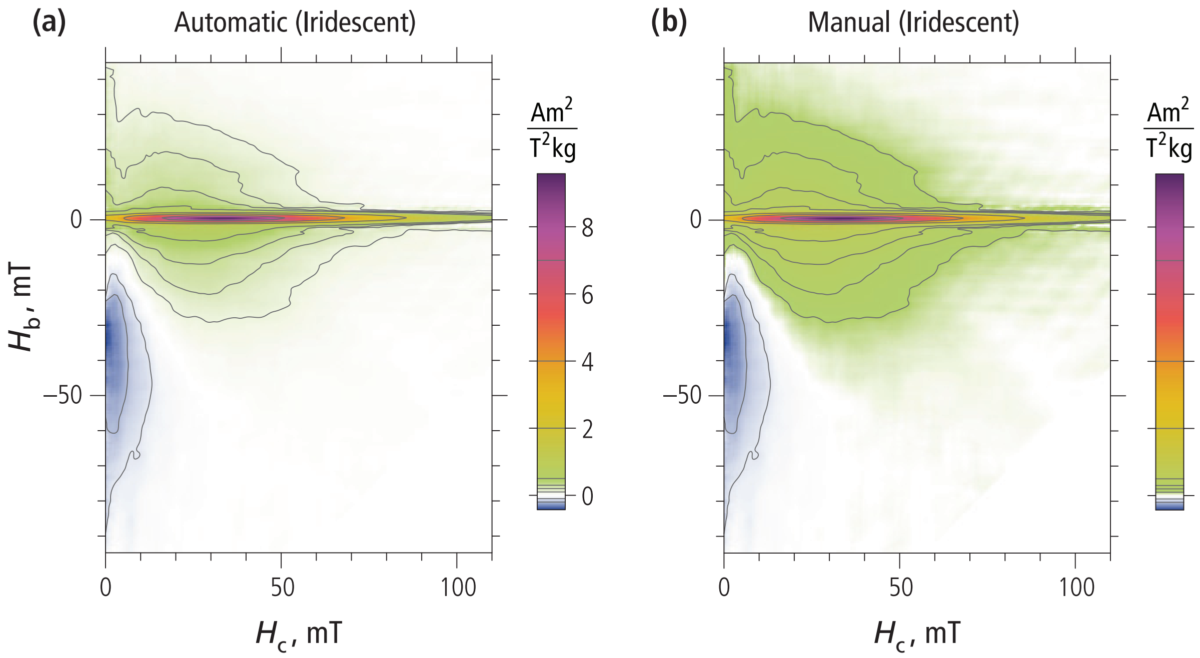


Fig. 5.18: `Iridescent` color scale examples based on the FORC diagram of a pelagic carbonate sample (see the downloadable example "pelagic carbonate"). **(a)** `INPUT 07` was set to `Automatic`. Negative FORC amplitudes are small but significant; therefore, `PlotFORC` chooses the expanded color range (blue-white-green) to coincide with the FORC range over negative values, and to be symmetric about 0. This means that blue, white, and green pixels correspond to ρ_{min}, 0, and $-\rho_{min}$, where ρ_{min} is the (negative) lower limit of the FORC diagram. **(b)** `INPUT 07` was set to `100`, so that the positive limit of the expanded color range (e.g. green) coincides with 100 mAm^2/T^2. Measurement noise in the lower quadrant of the FORC diagram becomes now visible. Because this part of the diagram is not significant, the green level set by `INPUT 07` is excessively small. The choice made with the `Automatic` setting in (a), i.e. green set at 429 mAm^2/T^2, gives best results.

The expanded color range can be also entered as percent values of the maximum FORC value, just by adding the % symbol at the end, e.g.:

```
INPUT 05. Expanded color range option ....; 10%
```

for an expanded range between ±10% of the maximum FORC value. This format, which is available since version 2.03, is particularly useful for comparing FORC diagrams through homogeneous plotting criteria. For example, the same level of color saturation for small values can be reached by setting the expanded color range to a fixed percent value, e.g. 10%.

It is highly recommended to set both `INPUT 05` and `INPUT 07` to `Automatic` for producing an initial test plot. In this case, `PlotFORC` tries to find parameters that are best suited to the chosen color scale and FORC diagram. These parameters are printed in a summary table and can be adapted to special requirements in a second run.

INPUT 08. FORC diagram ticks specification

`INPUT 08` is used to set the ticks of the H_c- and H_b-field scales in the FORC diagram. The default option is

```
INPUT 08. FORC plot ticks specifications ....; Automatic
```

for placing the field ticks automatically on the FORC diagram. In this case, `CalculateFORC` uses the same tick intervals (e.g. 50 mT for major ticks and 10 mT for minor ticks) for both H_c- and H_b-axes, as customary in FORC plots (Fig. 5.19a). In some cases, you might want to customize field ticks. Explicit tick specifications can be given with `INPUT 08` in two ways. The simplest specification consists of a couple of numbers, e.g.:

```
INPUT 08. FORC plot ticks specifications ....; 10, 2
```

where the first number is the field interval used for major ticks, and the second number is an integer >0 that specify the number of minor intervals by which major tick intervals are divided. The major tick interval is assumed to have the same unit as magnetic field specifications in the FORC matrix file (e.g. mT). The ticks specification in the above example means that major ticks are drawn every 10 mT, and that the 10 mT intervals are divided in two with a single minor tick. This specification applies to both H_c and H_b. Set the second number to 1 if you want to avoid minor ticks.

Different tick specifications for H_c and H_b, as for instance in case of FORC spaces with very different H_c- and H_b-ranges, can be entered with `INPUT 08` as well. In this case, four parameters are needed, e.g.:

```
INPUT 08. FORC plot ticks specifications ....; 5, 5, 50, 5
```

(Fig. 5.19b). The first and second number pair define H_c- and H_b-ticks in the same manner as with a single tick specification for both axes. In the above example, major H_c-ticks are drawn every 5 mT (first number), and major H_b-ticks are drawn every 50 mT (third number). Major H_c- and H_b-tick intervals are divided into five intervals (second and fourth number) by means of four minor ticks.

- The default tick specification `Automatic` generates satisfactory results in all cases where the FORC space is defined by similar H_c- and H_b-ranges.
- Explicit tick specifications are useful when dealing with FORC diagrams that are much more extended along one coordinate then the other (e.g. H_b in the case of multidomain samples).

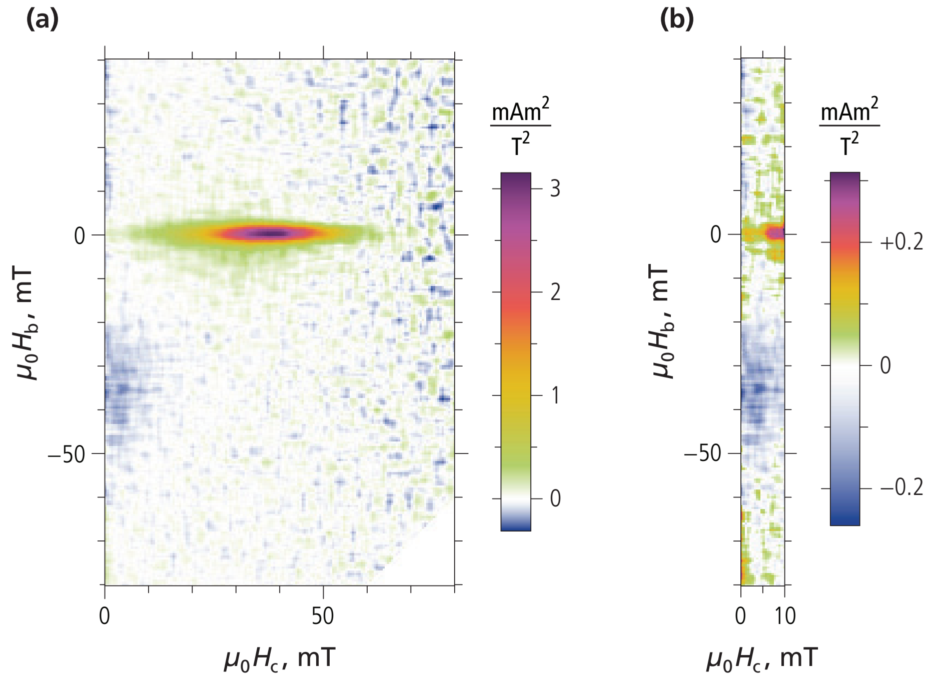


Fig. 5.19: FORC diagram examples obtained from same measurements of cultured magnetotactic bacteria (see the downloadable example "Magnetospirillum 2") over different field ranges. The field ticks in **(a)** have been generated with the default `Automatic` option for `INPUT 08`. The same result would be obtained by setting `INPUT 08` to `50,5` (i.e. major ticks every 50 mT with major tick intervals divided into 5 sub-intervals by minor ticks). The ticks in **(b)** have been specified by setting `INPUT 08` to `10,2,50,5`.

INPUT 09. Vertical exaggeration factor

FORC diagrams are usually plotted with their natural aspect ratio, which means that identical horizontal and vertical distances correspond to same H_c- and H_b-differences. The aspect ratio of FORC diagrams generated with `PlotFORC` is controlled with `INPUT 09` by entering is a positive factor representing the vertical stretch of a FORC diagram with natural aspect ratio. Preservation of the natural aspect ratio is therefore obtained by setting

```
INPUT 09. Vertical exaggeration factor ....; 1
```

(Fig. 5.20a). In some cases, it might be useful to stretch the FORC diagram vertically or horizontally in order to obtain a better representation of one-dimensional FORC features such as central ridges and vertical ridges. The FORC diagram is stretched vertically by entering a number >1, e.g.

```
INPUT 09. Vertical exaggeration factor ....; 5
```

for the example of Fig. 5.20c. Horizontal stretching by a factor *n*, on the other hand, is obtained by entering the reciprocal value 1/*n*. For example, the FORC diagram is stretched by a factor 5 along the horizontal axis upon entering

```
INPUT 09. Vertical exaggeration factor ....; 0.2
```

(Fig. 5.20e).

If the natural aspect ratio of the FORC diagram is drastically modified, it might be necessary to modify ticks specifications accordingly (see `INPUT 08`), in order to obtain a meaningful labeling of the stretched axis. For example, major H_b-ticks in Fig. 5.20c are set every 5 mT, instead of 50 mT, as for H_c. The color bar represented on the right side of the FORC diagram might also need to be rescaled for balanced appearance (see `INPUT 10`).

- Alteration of the natural aspect ratio of FORC diagrams can be misleading, and should be explicitly mentioned in figure captions.

Alteration of the natural aspect ratio of FORC diagrams is useful for highlighting details that would otherwise not be noticed. FORC example, the vertical shift of the central ridge in Fig. 5.20 becomes evident through a fivefold vertical exaggeration of the FORC diagram.

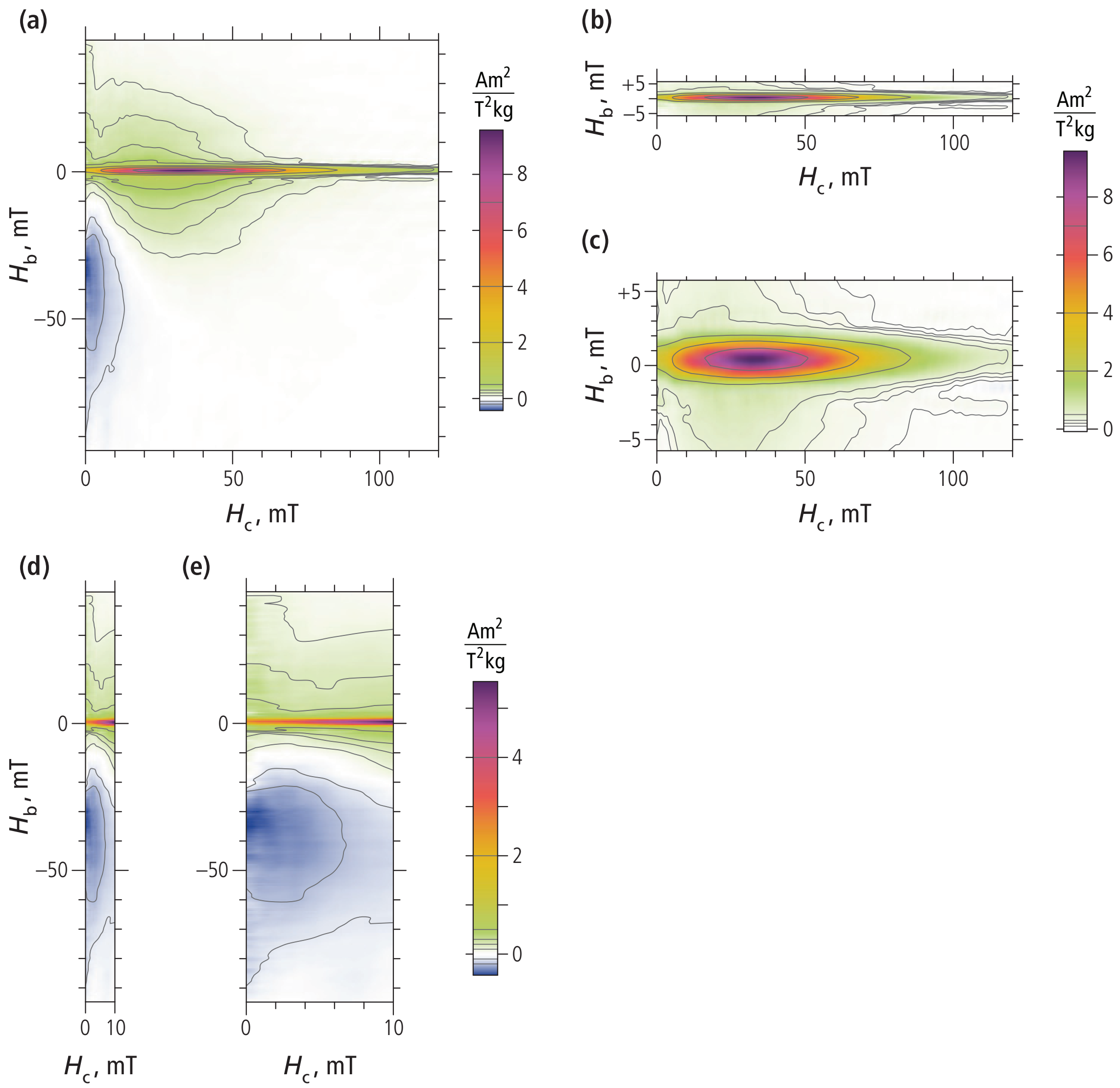


Fig. 5.20: Examples of FORC diagrams with modified aspect ratios generated from measurements of a pelagic carbonate (see the downloadable example "pelagic carbonate"). **(a)** FORC diagram covering the whole measured FORC space, with unmodified aspect ratio (i.e. `INPUT 09` set to `1`). **(b)** Same as in (a) after selecting a small area around the central ridge (i.e. `INPUT 02` set to `−6,6`). **(c)** Same as in (b), after stretching the diagram vertically by a factor 5 (i.e. `INPUT 09` set to `5`). The ~0.5 mT vertical shift of the central ridge is now clearly visible. **(d)** Same as in (a) after limiting the plotted range to small H_c -values (i.e. `INPUT 01` set to `0,10`). **(e)** Same as in (d), after stretching the diagram horizontally by a factor 5 (i.e. `INPUT 09` set to `0.2`).

INPUT 10. Color bar range

This is a new option available since version 3. It enables to choose the range of FORC values covered by the color bar. Two options are available: with the first option

```
INPUT 10. Color bar range .........; All
```

the color bar extends over the full range, regardless of how this is represented by the color scale (Fig. 5.21a). This option is recommended for most cases; however, there are situations where the FORC range includes artifacts with large positive or negative amplitudes that are of no interest. In this case,

```
INPUT 10. Color bar range .........; Automatic
```

can be used make the color bar coincide with the interval of FORC values that is covered by the color scale range chosen with `INPUT 05` (Fig. 5.21b). Appropriate choices of `INPUT 05` can then be used to exclude large artifact amplitudes from the color bar.

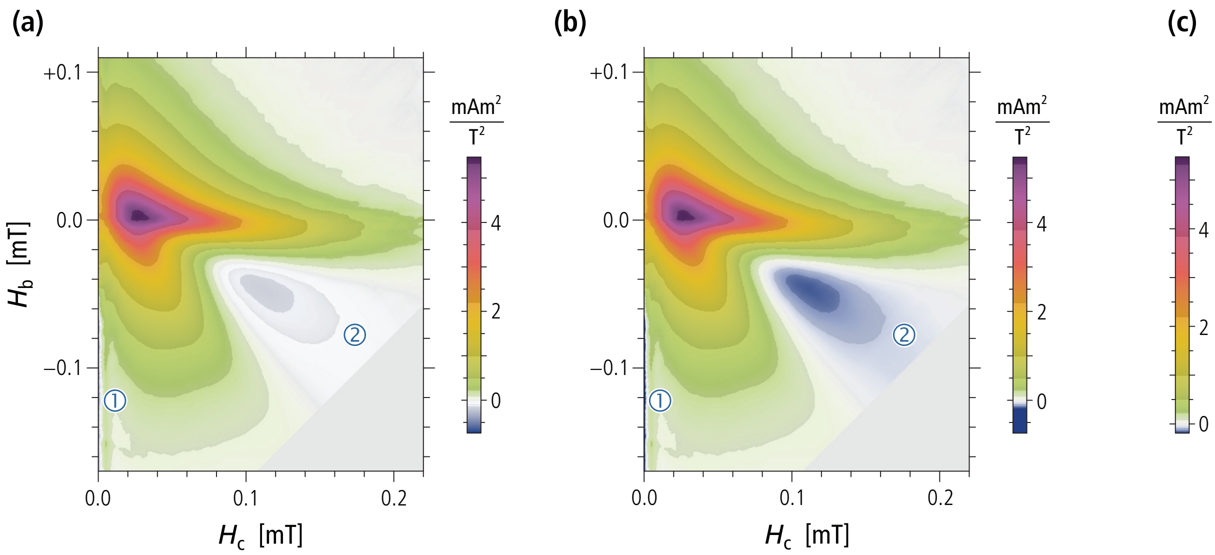


Fig. 5.21: **(a)** FORC diagram of a basalt sample with color scale extending over the whole range of FORC values (i.e. `INPUT 5` set to `All`). Negative amplitudes just above the $H_b = -H_c$ diagonal (marked with ②) are masked by the much larger negative ridge along $H_c = 0$ in the lower quadrant (marked with ①), which are a measurement artifact related to the first measurement point of each curve. **(b)** Same as (a), after limiting the lower end of the color scale (i.e. `INPUT 5` set to `-3.5,100%`). Negative amplitudes in ② are now clearly distinguishable. On the other hand, the color bar still includes the strong negative values corresponding to ① if the color bar range was set to `All` with `INPUT 10`, as it's the case here and in (a). **(c)** Color bar range obtained by setting `INPUT 10` to `All`: in this case, the lower end is controlled by the color range set with `INPUT 5`, i.e. −3.5% of the maximum value of the FORC function.

INPUT 11. Color bar normalization

Two options are available for labeling the color bar: Using the first option

```
INPUT 11. Color bar normalization .........; No
```

FORC amplitudes are indicated with their original unit (Fig. 5.22a). However, in many cases, absolute units are of no interest or difficult to interpret, so that everything is conveniently normalized by the maximum value of the FORC function that is covered by the unsaturated color scale range. This is obtained by setting

```
INPUT 11. Color bar normalization .........; Yes
```

In this case, FORC amplitudes are represented in percent (Fig. 5.22b), regardless of the original units. If the color scale range chosen with `INPUT 05` meets or exceeds the range of FORC values, normalization is performed with respect to the maximum of the FORC function. If, on the other hand, the color scale range is chosen to extend less than the maximum of the FORC function (for instance because this maximum is not significant), normalization is performed with respect to the upper limit of the color range, which is assumed to represent the "true" maximum amplitude of the FORC function.

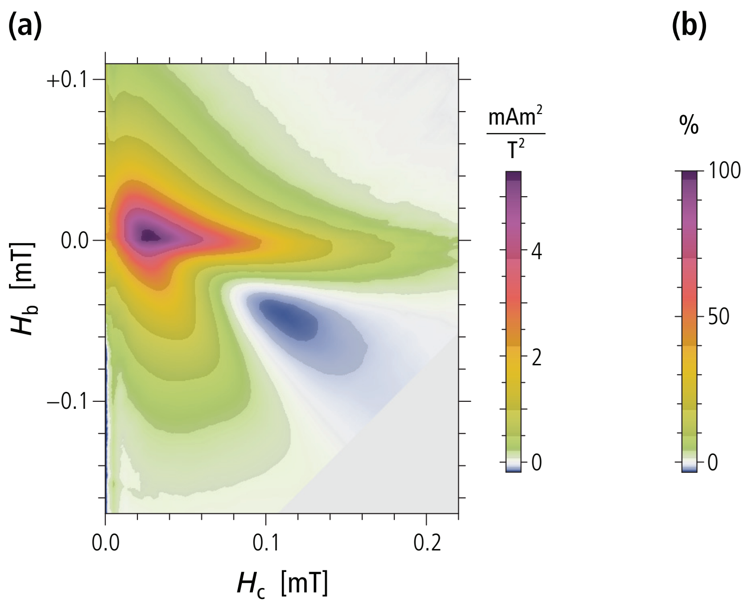


Fig. 5.22: (a) FORC diagram of a basalt sample with color bar expressed in original units of magnetic moment per magnetic field squared (i.e. `INPUT 11` set to `No`). **(b)** A color bar with FORC amplitudes expressed in percent of the maximum value is obtained by setting `INPUT 11` set to `Yes`.

INPUT 12. Color bar ticks

Color bar ticks are placed automatically when choosing

```
INPUT 12. Color bar ticks .........; Automatic
```

(Fig. 5.23a). The automatic option produces satisfactory results by evenly dividing the range of FORC values with 2-5 major ticks. Sometimes it might be more convenient to place major ticks in correspondence of contour levels, if contours are drawn. Direct labeling of the contours is obtained with

```
INPUT 12. Color bar ticks .........; Contours
```

In this case, all contour levels are labelled, except those that are very close to zero so to avoid overlapped labels (Fig. 5.23b). This option works only if contour drawing has been activated with `INPUT 14`; otherwise, the choice of color bar ticks is automatically reverted to `Automatic`. Complete color bar customization is obtained by entering an explicit list of FORC amplitudes to be labelled, e.g.

```
INPUT 12. Color bar ticks .........; 0,5,10,20,40,80%
```

for the example of Fig. 5.23c. As with contour levels and other FORC-amplitude-related parameters, numerical values entered with `INPUT 12` can be expressed in original units, or, as in this example, as percent of the maximum plotted amplitude. If percent values are to be entered, the list must end with the `%` symbol. The way `INPUT 12` is specified does not affect the unit used to express color bar labels. For example, the percent labels chosen in the example Fig. 5.23c are expressed through the corresponding values in original units if `INPUT 11` was set to `No` (Fig. 5.23d).

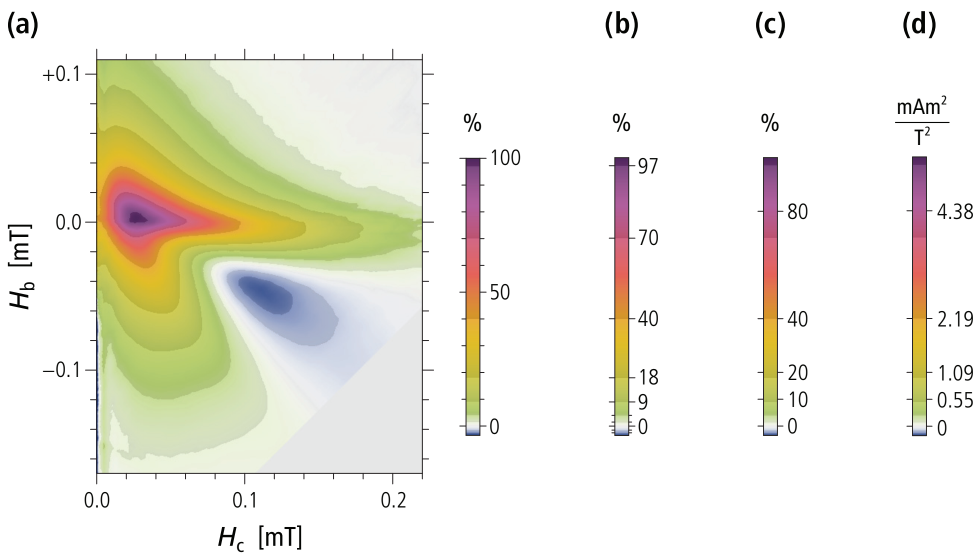


Fig. 5.23: (a) FORC diagram of a basalt sample with automatic color bar ticks (i.e. `INPUT 12` is `Automatic`). **(b)** Same as (a), after setting `INPUT 12` to `Contours` so that each contour is marked by a tick and labelled if there is enough spacing between successive contour levels. **(c)** Same as (a), after entering a list of explicit values (i.e. `INPUT 12` is `0,10,20,40,80%`). **(d)** Same as (c), when the color bar is expressed in original units (i.e. `INPUT 12` is `No`).

INPUT 13. Color bar vertical stretch

A color bar is placed by `PlotFORC` to the right of FORC diagrams. The color bar has default dimensions that match the diagram size. In some cases, especially with FORC diagrams that have very large or very small aspect rations, the color bar might be rescaled for best appearance. This is done with `INPUT 13`, which is defined as the numerical factor by which the color bar is stretched vertically with respect to its default appearance. Accordingly,

```
INPUT 13. Color bar vertical stretch ........; 1
```

gives the default color bar (Fig. 5.24a). Factors >1 can be entered to make the color bar taller, while the opposite effect is obtained with factors <1, e.g.

```
INPUT 13. Color bar vertical stretch ........; 0.5
```

for halving its height (Fig. 5.24b). `INPUT 13` has a purely esthetical function, since it does not modify the color scale content.

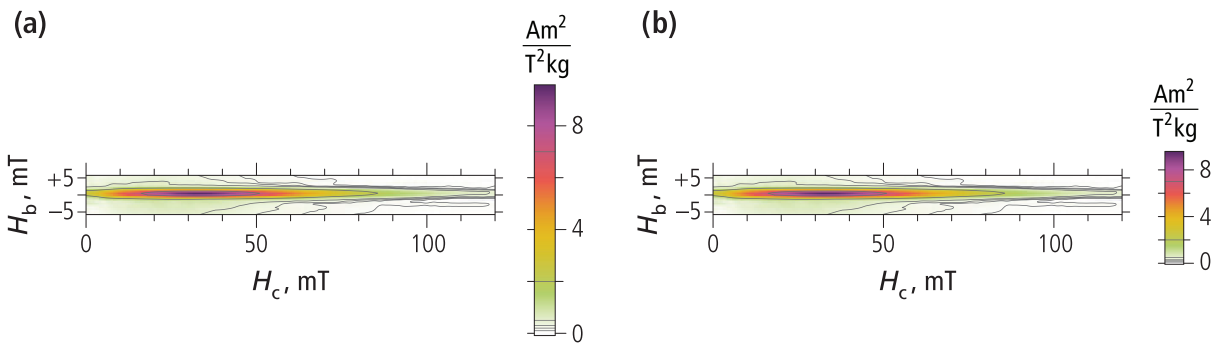


Fig. 5.24: FORC diagrams focusing on the central ridge of a pelagic carbonate (see the downloadable example “pelagic carbonate”). **(a)** FORC diagram and default color bar (i.e. `INPUT 13` was set to `1`). **(b)** FORC diagram with vertically shortened color bar, obtained with `INPUT 13` set to `0.5`.

INPUT 14. Contour level specification

Contours add visual information to the FORC diagram, highlighting features that are not always evident with color mapping alone (Fig. 5.25a,c). The main obstacle to FORC contour drawing is represented by measurement noise: in case of conventional processing of weak samples, meaningful contours can only be drawn around large-amplitude features that are already evident with color mapping (Fig. 5.25b,d). The advanced FORC processing options offered by `CalculateFORC` improve this situation significantly, because of much the higher signal-to-noise ratios attainable over regions of the FORC diagram with weakest amplitudes.

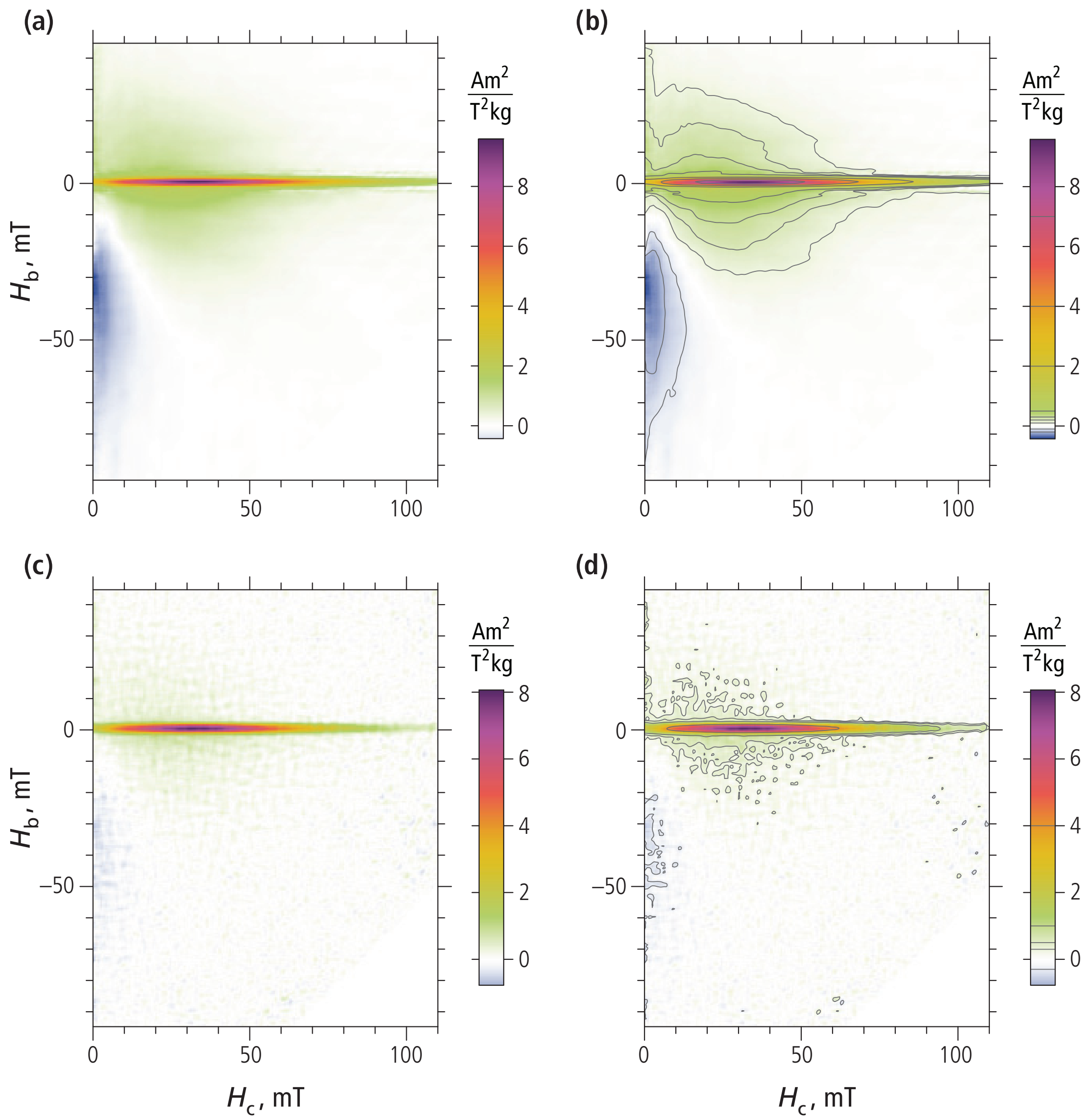


Fig. 5.25: Examples of FORC diagrams obtained from measurements of a pelagic carbonate (see the downloadable example "pelagic carbonate"). **(a,b)** FORC diagrams processed by `CalculateFORC` with advanced smoothing options, with and without contours. **(c,d)** FORC diagrams obtained with conventional processing (SF = 5), with and without contours. The diagram in (d) shows the elevated noise sensitivity of contours.

`PlotFORC` offers advanced contour drawing options controlled by `INPUT 14-17`. The first option, `INPUT 14`, contains specifications about the levels at which contours should be drawn. No contour is drawn with

```
INPUT 14. Contour specifications ....; None
```

(Fig. 5.25a). This option is always recommended for a first `PlotFORC` run, when other parameters, such as FORC range and color scale, need to be adjusted first. Contour levels can be specified in two manners. The simplest one consists in entering the number of contours to be drawn, i.e. an integer >1. For example, 10 contours are drawn with

```
INPUT 14. Contour specifications ....; 10
```

In this case, `PlotFORC` finds 10 equally spaced contour levels over the whole range of FORC values, taking only significant amplitudes into account. This option works relatively well with regular FORC functions (Fig. 5.26a). However, best results are generally obtained with manually chosen contour levels (Fig. 5.26b).

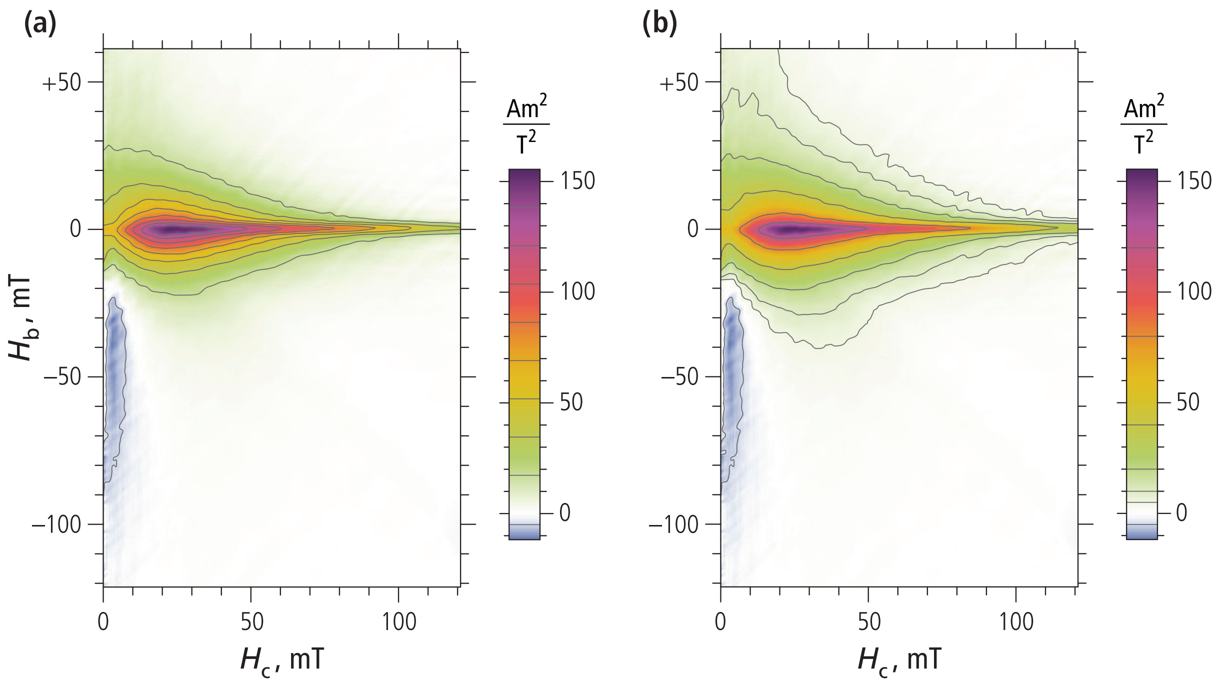


Fig. 5.26: Contour drawing examples based on a FORC diagram of extracted fossil magnetosomes (see the downloadable example "extracted magnetosomes"). **(a)** Automatic drawing of 8 contours (i.e. `INPUT 14` set to `8`). As shown in this example, equally spaced contour levels tend to produce excessive clustering near the central peak. Furthermore, weak, but significant contribution far from the central peak remain uncovered. **(b)** Contours at −5, 5, 10, 20, 40, 80, and 120 Am^2/T^2, specified explicitly with a list of contour levels (i.e. `INPUT 14` set to `-5,5,10,20,40,80,120`).

If significant negative amplitudes are found, `PlotFORC` tries to include at least one negative contour level by defining a "high-resolution range" over absolute FORC amplitudes comprised between 0 and the absolute value of the largest negative amplitude. This process is similar to the definition of an expanded color range (see `INPUT 07`), in which contour levels are more closely spaced (i.e., all `Iridescent` colors between blue and green, and all `Sunsky` colors between blue and yellow). An example is shown in Fig. 5.26a, where the negative contour level is closer to zero than other levels.

In general, the automatic generation of a fixed number of contours does not perform optimally with FORC functions that contain localized, high-amplitude features, such as the central ridge (Fig. 5.27a). In this case, more contours are drawn over high-amplitude features than necessary. This problem is avoided by entering a list of appropriately chosen contour levels, e.g.

```
INPUT 14. Contour specifications ....; -200,-100,100,200,300,500,2000,4000,7000
```

for the example of Fig. 5.27b. All contour levels must be entered in a single line and separated by commas. The entered numbers are understood as FORC amplitudes expressed in original FORC matrix units.

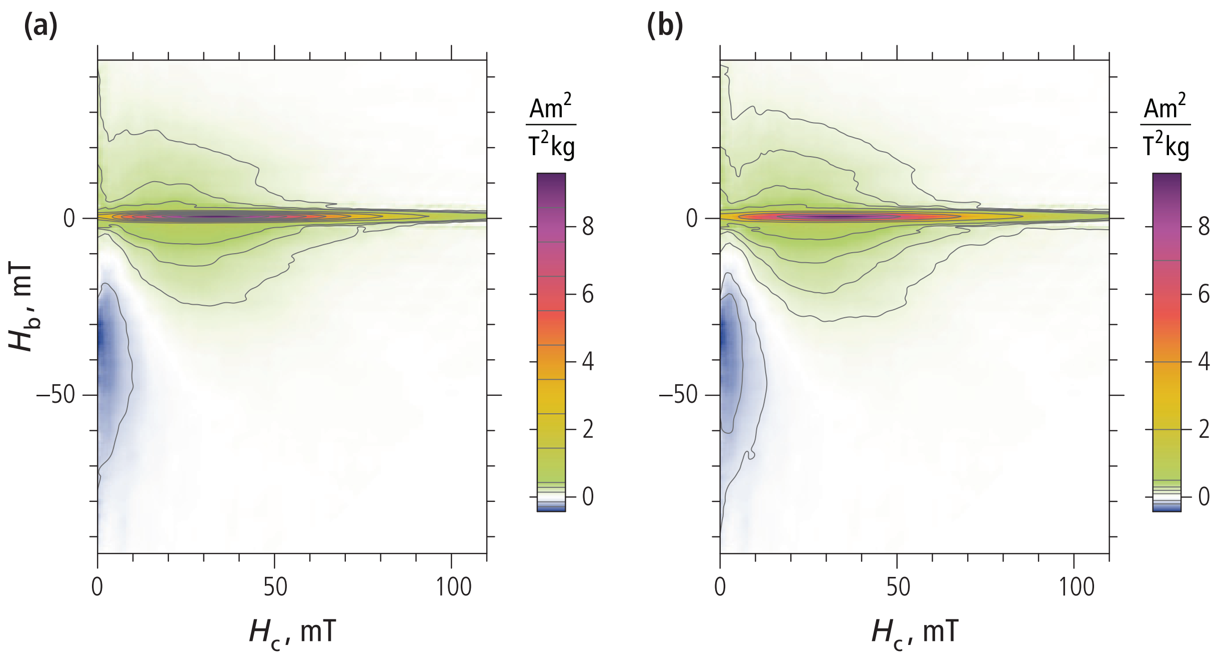


Fig. 5.27: Contour drawing examples based on the FORC diagram of a pelagic carbonate sample (see the downloadable example "pelagic carbonate"). **(a)** Automatic drawing of ~10 contours at equally spaced levels (i.e. `INPUT 14` set to `10`). Notice the excessive number of contours over the central ridge. **(b)** Contours at ±0.2, ±0.1, 0.3, 0.5, 2, 4, and 7 $Am^2/(T^2\ kg)$ specified by setting `INPUT 14` to `−200,−100,100,200,500,2000,40000,7000`, according to the original FORC matrix unit $mAm^2/(T^2\ kg)$.

`PlotFORC` provides a summary table of FORC and color scale ranges, expressed in original units, e.g.

```
Range parameters for color scale Iridescent in original units:
FORC range .........: from -429.391 ±100.143 to 9580.5 ±210.068
Color scale range ..: from -429.391          to 9580.5
Green at ...........: 100
```

which helps with a definition of a suitable list of contour levels. In the (rare) eventuality of entering a single contour level, ensure yourself that you do not enter an integer number, which would be mistakenly interpreted as the number of contours to draw at automatically chosen levels. For example,

```
INPUT 11. Contour specifications ....; 200.0
```

is entered for drawing a single contour at 200 $mAm^2/(T^2 kg)$. In order to avoid the consequences of accidental confusion between a single contour level specification and contour number, the maximum number of contours that can be drawn is limited to 100.

For best results, contour levels should be specified with the following procedure in mind:

1) Make a first `PlotFORC` run without contours to obtain diagnostic data, such as the range of FORC values and the corresponding standard errors.

2) Enter the smallest positive contour level that is not excessively affected by measurement noise. Noise produces meandering contours that should be avoided. For the same reason, never set a contour level to zero.

2) Once the first contour is drawn, choose a second, larger contour level so that the distance between the two contours in the FORC diagram (not the color bar!) matches your requirements. Avoid excessively close contours, which give the FORC diagram a crowded appearance without adding clarity.

3) Continue to add contour levels with the same criteria, until you reach maximum amplitudes (Fig. 5.26a), or amplitudes corresponding to the beginning of high-amplitude, low-dimensional FORC features (e.g. the central ridge, Fig. 5.27b). Contours should be regularly spaced in the FORC diagram, but not necessarily in the color bar.

4) If the FORC diagram contains significant negative amplitudes, add symmetric negative contour levels (i.e. the same numbers as positive contour levels, but with reversed sign). Negative amplitudes are usually small, and will therefore be covered only by 1-2 contours (Fig. 5.27b).

5) If the FORC diagram contains high-amplitude, low-dimensional features, such as the central ridge, choose few contour levels to cover the corresponding amplitudes. Because of the limited extension of such features, only few contours should be drawn over them (Fig. 5.27b).

Best contouring results are obtained with explicit contour level specifications.

- Contours should be drawn only at levels for which the FORC function is significant. See `INPUT 18` for marking the FORC region containing significant amplitudes.

INPUT 15. Contour spline type

Contours are defined by spline lines with spline parameters controlled by INPUT 15. PlotFORC offers a choice between two types of splines: B-spline of polynomial degree *p*, and Bézier spline. The specification of a B-spline of polynomial degree *p*, is BSpline, *p*, e.g.

```
INPUT 15. Contour spline type ....; BSpline, 3
```

for third-order splines (Fig. 5.28a). Small-scale contour irregularities are smoothed with increasing polynomial degrees; therefore, large degrees (e.g. 100, Fig. 5.28c) are recommended. Short contours are calculated with the maximum possible polynomial degree that is compatible with the number of knot points.

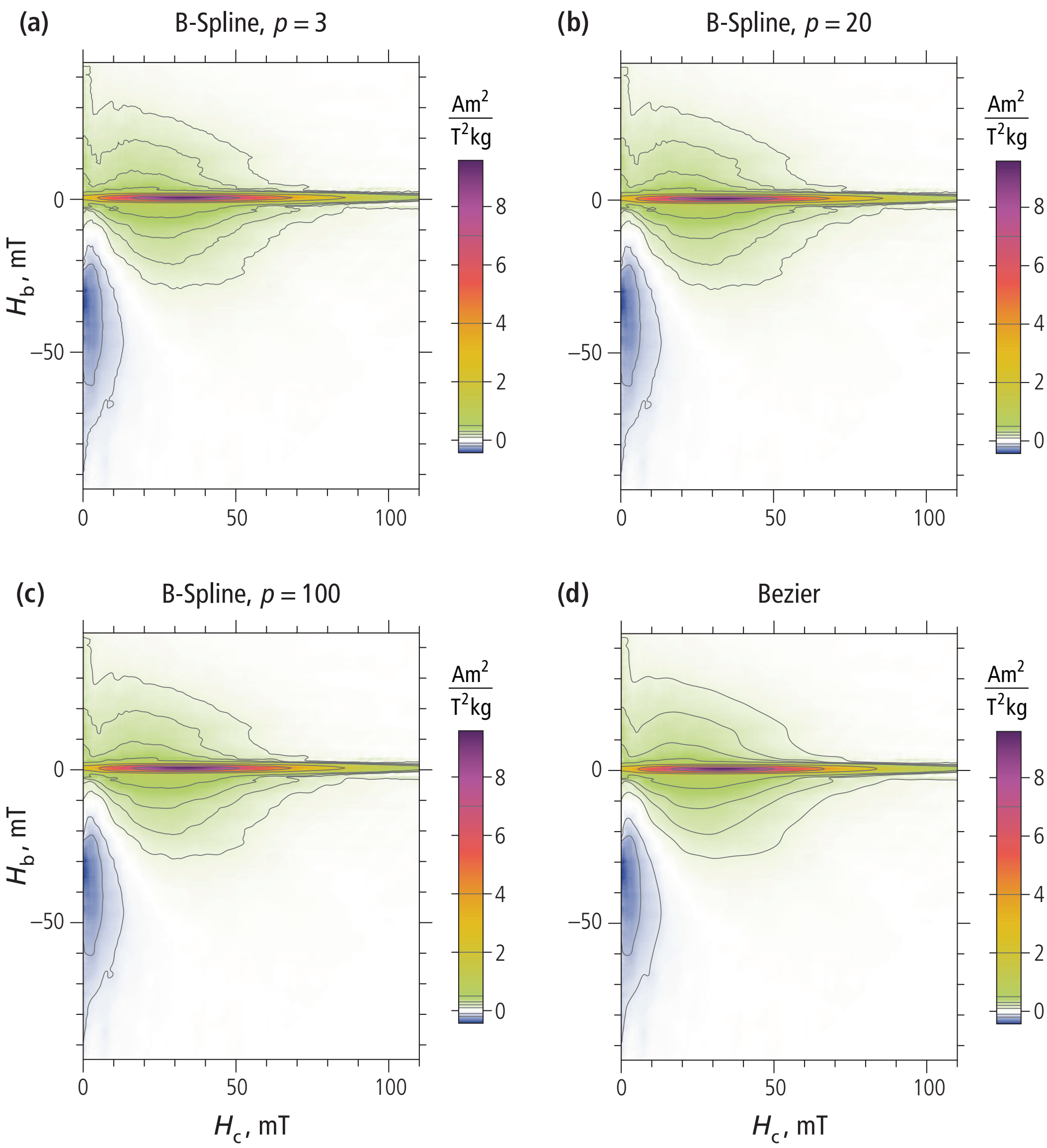


Fig. 5.28: Spline option examples based on FORC diagrams of a pelagic carbonate (see the downloadable example "pelagic carbonate"). **(a)** B-splines with degree $p = 3$ (i.e. INPUT 15 set to BSpline,3). **(b-c)** Same as in (a) for $p = 20$ and 100. **(d)** Bézier splines (i.e., INPUT 15 set to Bezier).

Bézier is a special case of splines with no internal knots, whose polynomial degree is automatically set by `PlotFORC`. Bézier splines are obtained with:

```
INPUT 15. Contour spline type ....; Bezier
```

(Fig. 5.28d). Bézier splines generally yield smoother (but more inaccurate) contours than B-splines.

- B-Splines with a maximum polynomial degree of 100 (i.e. `INPUT 15` set to `BSpline,100`) perform well in most circumstances and can be used as default setting.
- `INPUT 12` has no meaning if contours are not drawn (i.e. `INPUT 14` set to `None`).

- The choice of a contour spline type is irrelevant if the contour style option (`INPUT 17`) is set to a color band, i.e. `step` or `shade`, since in this case contours are rendered by color scale discontinuities, and not by lines.

INPUT 16. Exclusion limit for short contours

Measurement noise often produces small closed contours around local maxima of the FORC function, which are not significant. These contours yield crowded diagrams while not providing additional information, and should be avoided. Elimination of insignificant contours is controlled by a minimum contour length threshold entered with `INPUT 16`. Contour shorter than this threshold are not drawn. The threshold length is expressed as fraction of the longest contour in the FORC diagram: for example,

```
INPUT 16. Exclusion limit for short contours ; 0.2
```

makes `PlotFORC` exclude all contours whose length is <20% of that of the longest contour (Fig. 5.29a). On the other hand, all contours are drawn by setting

```
INPUT 16. Exclusion limit for short contours ; 0
```

(Fig. 5.29b). It is recommended to draw all contours in a first `PlotFORC` run, and to stepwise increase the contour length threshold until undesired small contour loops are eliminated.

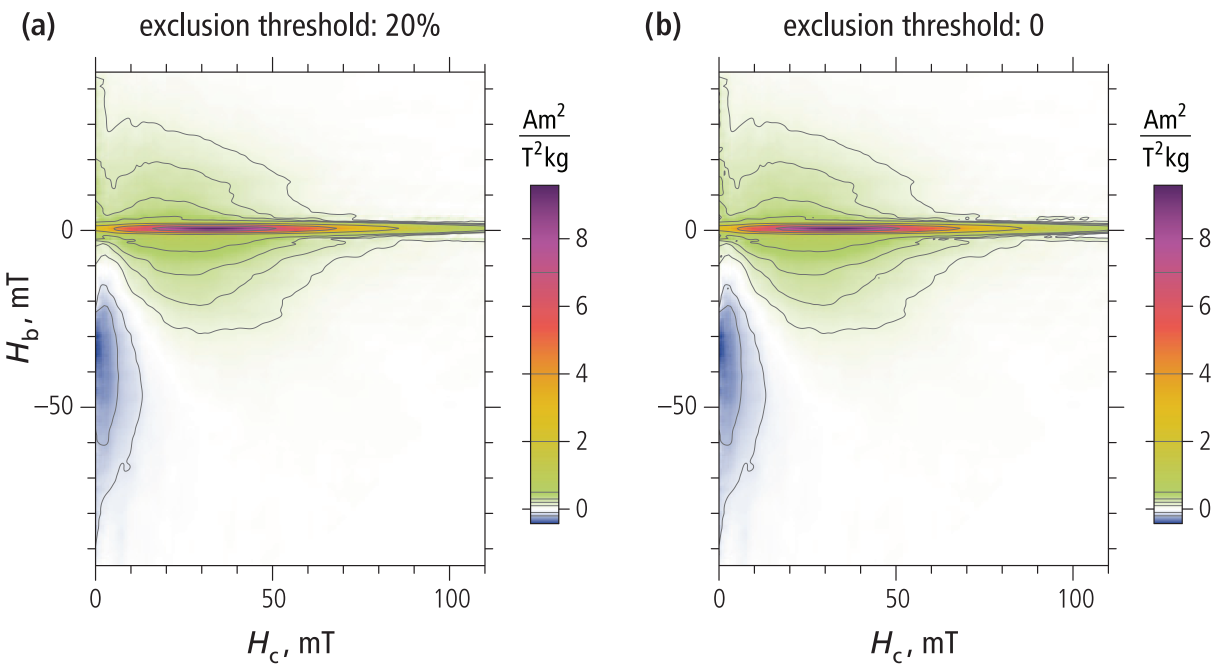


Fig. 5.29: Examples of contour length thresholds based on FORC diagrams of a pelagic carbonate (see the downloadable example “pelagic carbonate”). **(a)** Contours whose length is <20% of the longest contour line are not drawn in this diagram (i.e. `INPUT 16` set to `0.2`). **(b)** All contours are drawn with `INPUT 16` set to `0`. Notice the small contour loops above and below the central ridge, which are absent in (a).

- Whole contour levels might not be drawn if `INPUT 16` approaches `1`. Therefore, `INPUT 16` should be carefully increased in small steps, so that significant contours are retained.

INPUT 17. Contour style

The style of contours is set with INPUT 17, whereby the main choice is made between line contours and contour rendering through color scale discontinuities. In the case of line contours, INPUT 17 sets the line color, with possible color choices given in Table 5.1. The recommended contour color for `Iridescent` and `Sunsky` color scales is

```
INPUT 17. Contour style ....; gray
```

(Fig. 5.30a). On the other hand, black contours (i.e. INPUT 17 set to `black`) are best used in with the monochrome color scales `Aquamarine` and `Sepia` (Fig. 5.30b).

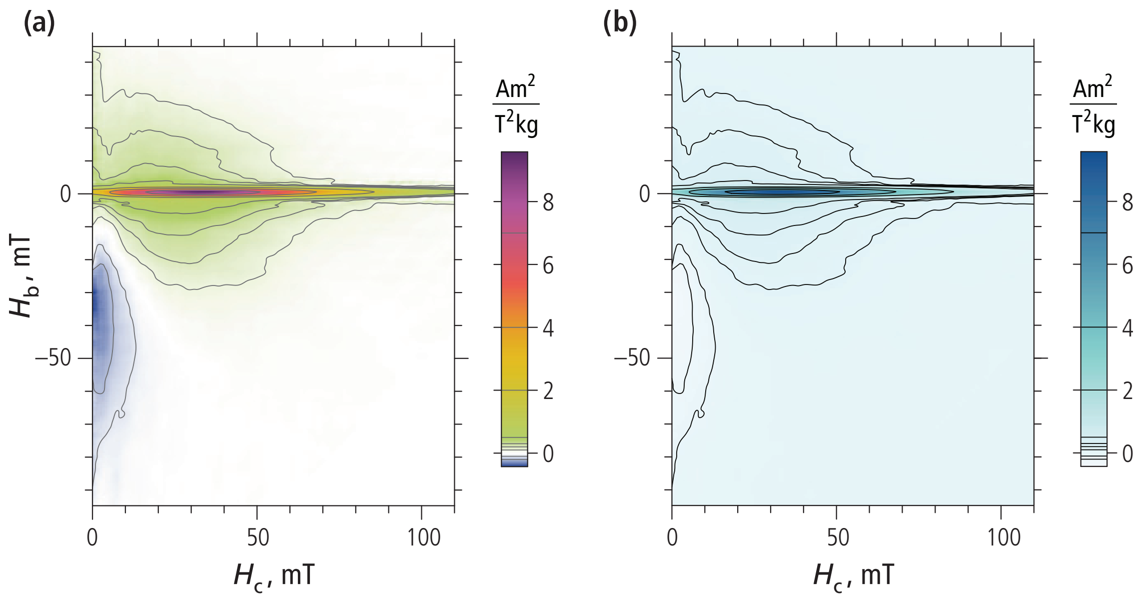


Fig. 5.30: Examples of contour drawing styles with FORC diagrams of a pelagic carbonate (see the downloadable example "pelagic carbonate"). **(a)** Gray contours with the `Iridescent` color scale. **(b)** Black contours with the `Aquamarine` color scale.

Table 5.1: List of contour color specifications and recommended usage.

`INPUT 14` color	Corresponding RGB color	Recommended for
`black`	`R = 0 , G = 0 , B = 0`	Color scales `Aquamarine`, `Sepia`
`gray`	`R = 76 , G = 76 , B = 76`	Color scales `Iridescent`, `Sunsky`
`white`	`R = 255, G = 255, B = 255`	—
`red`	`R = 255, G = 0 , B = 0`	Significance contour (see INPUT 15)
`blue`	`R = 0 , G = 0 , B = 127`	Color scales `Iridescent`, `Sepia`
`green`	`R = 0 , G = 127, B = 0`	Color scale `Sepia`

Additional contour style options have been added in v.2.3. With these options, contours are marked by slight discontinuities of the color scale. For this purpose, two options can be chosen: `step` and `shade`. In the first case, i.e.:

```
INPUT 17. Contour style ....; step
```

the color scale is discretized at the contour levels specified with `INPUT 14`, so that the space enclosed by any two contours is homogeneous in color (Fig. 5.31a). With the second option `shade`, the color scale discontinuities are enhanced by adding shades (Fig. 5.31b). These drawing options are strongly affected by the image resolution, that is, the dimension of pixels representing individual FORC values. Low-resolution images (e.g. Fig. 5.32a) acquire a grainy appearance, which can be mitigated to a certain extent by adding a small level of blur. Optional blur is specified by an additional number indicating the blur radius in pixels, e.g.:

```
INPUT 17. Contour style ....; step, 1.4
```

for a 1.4-pixel-blur yielding the diagram shown in Fig. 5.32b. This option can also be used with `shade`. Recommended blur radii are comprised between 1 and 2, although best results are always obtained by generating FORC diagrams with a sufficiently small mesh size (see `INPUT 06` for `Calculate FORC`).

Contour contrast generated by the `shade` option depend on the maximum amount of color darkening. The default value is 10% of saturation. Other values can be optionally entered after the blur value, e.g.

```
INPUT 17. Contour style ....; shade, 0, 0.2
```

for no blur and shades set to 20% of saturation.

When the contour style is set to `step` and `shade`, contour lines are automatically dropped and `INPUT 15` as well as `INPUT 16` become irrelevant. Due to the lack of contour smoothing and elimination of small contour loops, `step` and `shade` contour styles are not suited for the representation of FORC diagrams with evident noise contributions.

- Contour representation by color scale discontinuities (i.e. `INPUT 17` set to `step` or `shade`) is recommended only for FORC diagrams with >300 points along each direction. The amount of points can be freely chosen during FORC diagram calculation using the mesh size option (`INPUT 06`) of `Calculate FORC`.

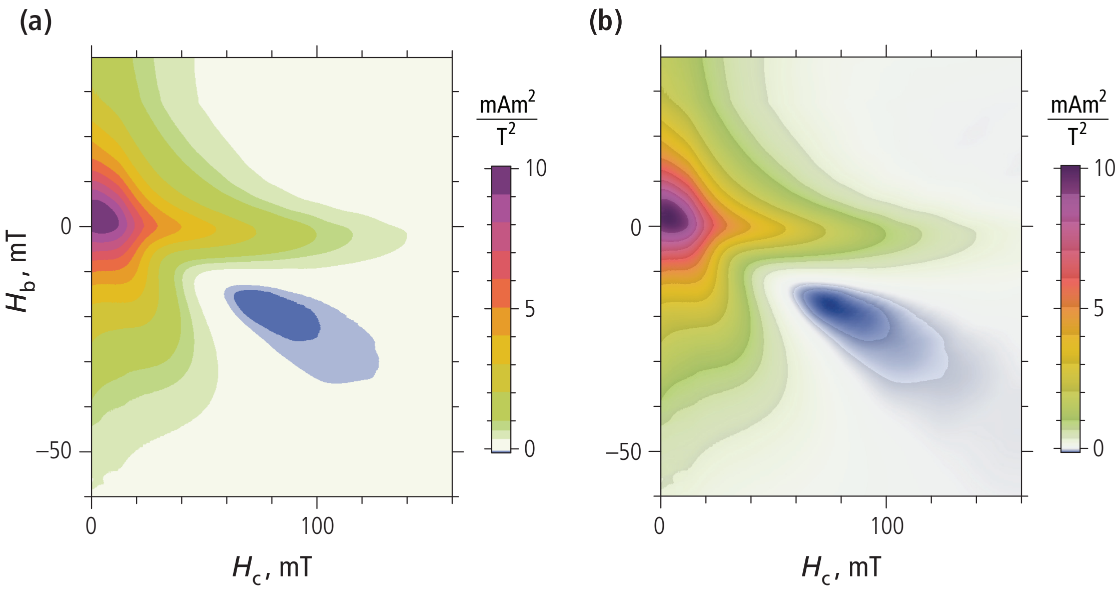


Fig. 5.31: Examples of contour drawing through discontinuous color scales, based on the FORC diagram of a basalt sample. Ten contour levels have been automatically generated by setting `INPUT 14` to `10`. **(a)** Contour style option `step`, and **(b)** contour style option `shade`, with no further specifications.

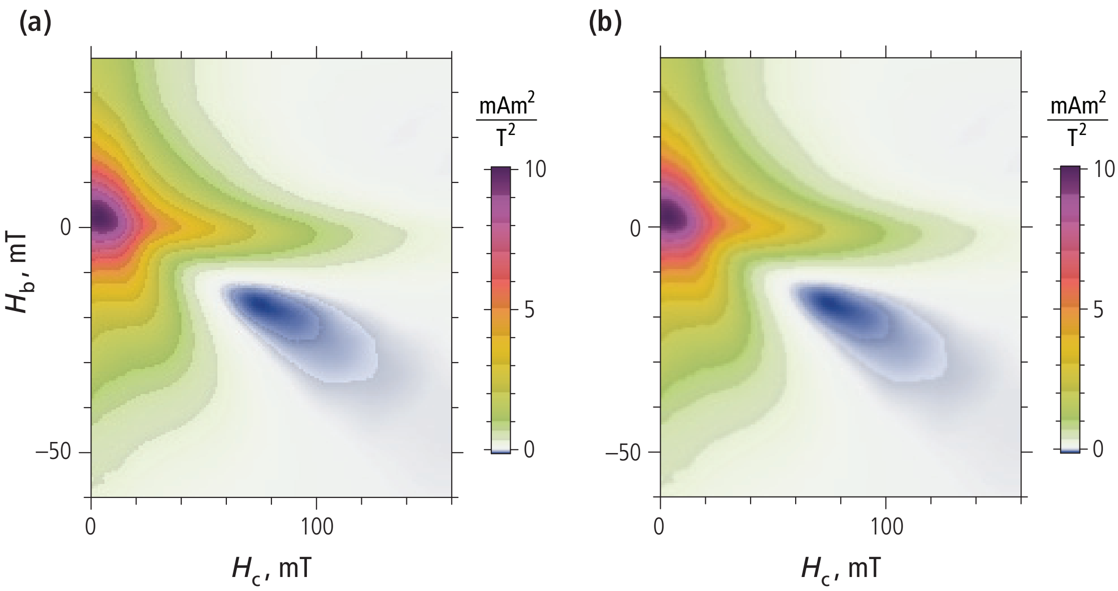


Fig. 5.32: Contour drawing through discontinuous color scales with a low-resolution FORC diagram of a basalt sample. Ten contour levels have been automatically generated by setting `INPUT 14` to `10`. **(a)** Contour style option `shade` with no further specifications: notice individual pixels along the color band boundaries. **(b)** Same as (a), after introducing a 1.4 pixel blur (i.e. `INPUT 17` set to `shade,1.4`).

INPUT 18. Significance contour specification

Often, only parts of the plotted FORC diagram contain significant amplitudes. `PlotFORC` offers the possibility to highlight significant regions of the FORC diagram according to the criteria of *Heslop and Roberts* [2012]. This is done by drawing specially marked contours around significant regions of the FORC diagram. This options works independently from regular contour drawing (i.e. `INPUT 147`), so that significant regions can be highlighted also if other contours are not drawn. Significant regions of the FORC diagram are left unmarked upon setting

```
INPUT 18. Significance contour level ........; None
```

Otherwise, the threshold for significant signal-to-noise ratios is entered with `INPUT 18` in form of a positive number >0, followed by `All` for drawing all contours corresponding to the given threshold (including small loops produced by measurement noise), or by the number of longest contours to be drawn (usually 1-2 for positive FORC functions and 3-4 if significant negative amplitudes are present). The significance threshold is defined as the signal-to-noise ratio above which the FORC function is ≠ 0 at a given confidence level, usually chosen to be 95% or 99% [*Heslop and Roberts*, 2012]. This threshold depends on the number of measurements taken for the calculation of the FORC function at a given point in FORC space. This number is >25 for most cases, so that the significance threshold corresponding to 99% confidence level is always equal or slightly smaller than 3 [*Egli*, 2013]. Accordingly, FORC values with signal-to-noise ratios ≥3 can be considered significant at a 99% confidence level. In this case

```
INPUT 18. Significance contour level ........; 3, All
```

is used to draw all contours corresponding to a significance threshold of 3 (Fig. 5.33a). Significance contours are drawn with thick lines and a color specified by `INPUT 19`. Significance thresholds corresponding to commonly used confidence levels are listed in Table 5.2.

- Significance contours can be used in preliminary runs of `PlotFORC` to ensure that contour lines are drawn only over significant regions of the FORC diagram.
- Significance contours are often nearly proportional to FORC amplitudes and therefore coincide with ordinary contour lines. In this case, regular contours can be used to delimit significant regions of the FORC diagram.
- The significance contour is particularly useful in cases where the signal-to-noise ratio is not proportional to FORC amplitudes. This is mostly true for FORC diagrams obtained with advanced smoothing options.

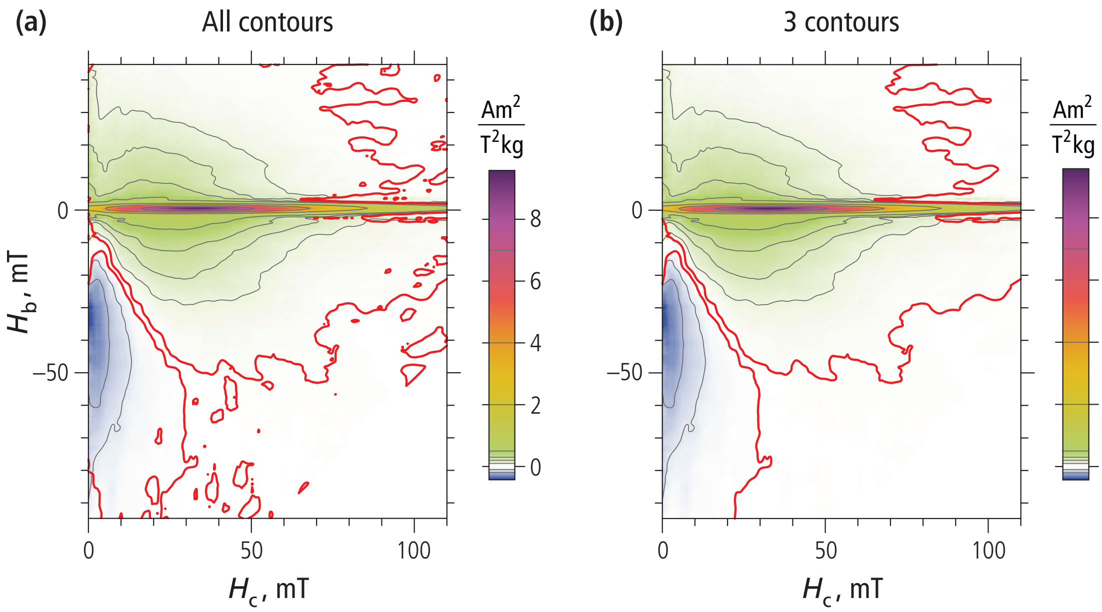


Fig. 5.33: Examples of significance contours (red) based on the FORC diagram of a pelagic carbonate (see the downloadable example "pelagic carbonate"). In both cases, the significance threshold was set to 3 (i.e. the first entry of INPUT 18 is 3), so that FORC regions included by red contours are ≠ 0 at 99% confidence level. **(a)** All significance contours are drawn (i.e. the second option of INPUT 18 is `All`) **(b)** Same as (a) after drawing only the 3 longest contours (i.e. the second option of INPUT 18 is 3).

As with regular contours, significance contours are affected by measurement noise, so that small contour loops with no meaning appear along with the major contours delimiting significant regions of the FORC diagram (Fig. 5.33a). In order to avoid such small loops, the number of longest significance contours to be drawn can be specified in INPUT 18 by a second number following the significance level, e.g.

```
INPUT 15. Significance contour level ........; 3, 3
```

for drawing the three longest contours shown in Fig. 5.33b.

- Meaningful significance contours can be selected from a first `PlotFORC` run where all contours are drawn (i.e. the second entry of INPUT 18 is `All`). The number of longest contours that is effectively needed (usually 1-2 for positive FORC functions, and 3-4 for FORC functions with negative contributions) is then entered with INPUT 18 for a second run.
- Significance contours are particularly useful during preliminary `PlotFORC` runs as a tool for ensuring that regular contour lines do not extend beyond significant regions of the FORC diagram.

Table 5.2: Significance thresholds of the signal-to-noise ratio for given confidence levels $1-\alpha$. These thresholds are defined as the $1-\alpha/2$ quantile of the Student *t*-distribution with ν degrees of freedom, where $\nu = 8(s_c-1)(s_b-1)+6(s_c+s_b)-13$ depends on the smoothing factors s_c along H_c and s_b along H_b used for FORC processing. Signal-to-noise ratios larger than the given significance threshold represent values of the FORC function that are ≠ 0 at a given confidence level.

	$s_c = s_b = 2$	$s_c = 5$, $s_b = 2$	$s_c = s_b = 10$
Degrees of freedom	19	61	755
99% confidence level	2.86	2.66	2.58
95% confidence level	2.09	2.00	1.96
90% confidence level	1.73	1.67	1.65

INPUT 19. Significance contour color

If regular contours are drawn together with the significance contour, the style of the latter should be clearly distinguishable from the other contours. The significance contour is plotted with a thicker line whose color is set by INPUT 19. Possible color specifications are listed in Table 5.3. Red, i.e.

```
INPUT 19. Significance contour color ........; red
```

provides the best contrast against the color scales `Iridescent` (Fig. 5.33) and `Aquamarine`, while green best suited for use with the color scales `Sunsky` and `Sepia` (Fig. 5.34).

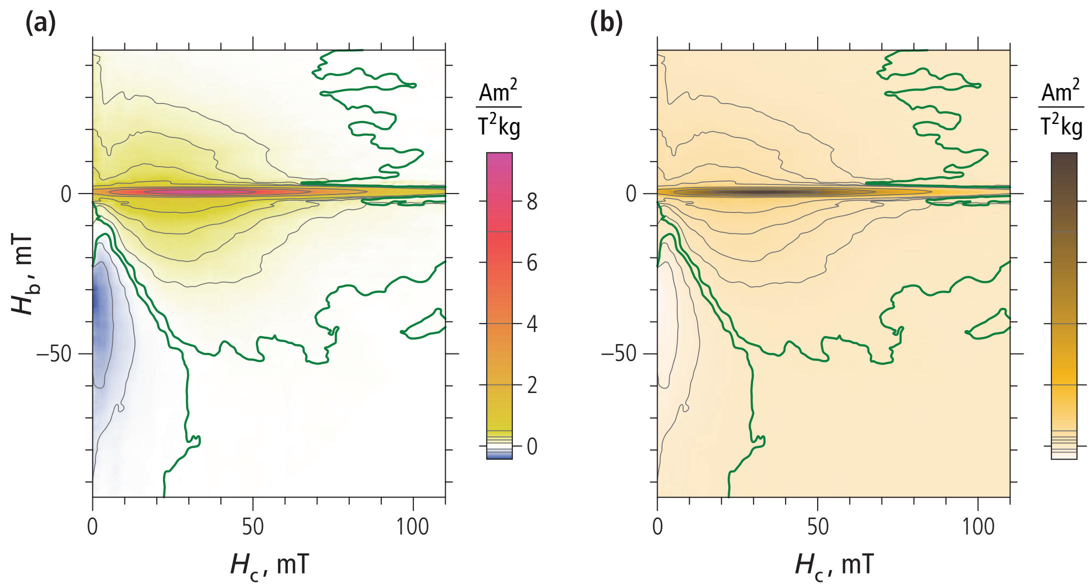


Fig. 5.34: Examples of significance contour styling (i.e. INPUT 19 set to `green`) based on the FORC diagram of a pelagic carbonate (see the downloadable example "pelagic carbonate"). **(a)** Significance contours drawn over a `Sunsky` color scale. **(b)** Significance contours drawn over a `Sepia` color scale.

Table 5.3: List of significance contour color specifications and recommended usage.

INPUT 16 color	Corresponding RGB color	Recommended for color scales
black	R = 0 , G = 0 , B = 0	Aquamarine, Sepia (no other contours)
gray	R = 76 , G = 76 , B = 76	Iridescent, Sunsky (no other contours)
white	R = 255, G = 255, B = 255	—
red	R = 255, G = 0 , B = 0	Iridescent, Aquamarine
blue	R = 0 , G = 0 , B = 127	Iridescent, Sepia (no other contours)
green	R = 0 , G = 127, B = 0	Sunsky, Sepia

INPUT 20. FORC diagram export format

The FORC diagram produced by `PlotFORC` is exported in a graphics format, unless `INPUT 20` is set to `None`, i.e.

```
INPUT 20. FORC diagram export format ....; None
```

Otherwise, `INPUT 20` is set to one of the following two vector graphics formats: `EPS` for encapsulated post script, and `PDF` for Adobe's portable document format, or one of the following two raster formats: `JPEG` (print) and `GIF` (web usage). For example,

```
INPUT 20. FORC diagram export format ....; PDF
```

is used to export the FORC diagram as a PDF file. In case a valid graphics format is entered with `INPUT 20`, a destination file for the FORC graphics must be chosen though a "save file" dialog opened by `PlotFORC`.

FORC graphics saved with the abovementioned formats have a similar appearance as the examples shown in this manual. Vector graphics are highly recommended over raster formats.

- Vector graphics are the recommended export format, since they allow post processing adjustments of labels, frame, ticks, and contours. This is particularly important for the preparation of printed matter (articles, book chapters, etc.).
- The color map of the FORC diagram is always saved as raster image, even in case of vector graphics. The resolution of the color map corresponds to the original resolution of the FORC matrix, so that this representation is lossless while avoiding complications and excessive memory usage related to vector representations of color pixels. The remaining elements of the FORC diagram (e.g. frame, labels, and contours) are saved as vectors if required by the output format.
- Raster graphics are produced with large enough resolution for most purposes. If a higher resolution is desired, export graphics in vector format and convert them with image processing software.

- Font problems can arise with vector graphics. In this case, font styling (e.g. italic *H* in frame labels) might be lost, and special fonts (e.g. the negative sign of numbers) might disappear or display incorrectly. These problems can be solved with image processing software (e.g. Adobe Illustrator® and Corel Draw®). If `PlotFORC` is run with Mathematica® 10, fonts are automatically converted to outlines, eliminating font coding problems. On the other hand, post-processing font manipulation in vector graphics is no longer possible.

INPUT 21. Vertical profile normalization

The interpretation of FORC diagrams is often based on their identification with the Preisach function

$$p(H_c,H_b)=f(H_c)g(H_b) \tag{5.1}$$

which represents the probability density of elemental rectangular hysteresis loops (so-called hysterons) in terms of a distribution $f(H_c)$ of coercive fields and a distribution $g(H_b)$ of bias fields H_b by which the hysterons are shifted horizontally [*Preisach*, 1935]. In rare cases, hysterons can be associated with the hysteresis of individual, single-domain particles [*Néel*, 1958], so that $f(H_c)$ and $g(H_b)$ are the distributions of particle coercivities and local interaction fields, respectively. Accordingly, normalization of the Preisach function by its value at $H_b=0$ yields a new function

$$p^*(H_c,H_b)=\frac{g(H_b)}{g(0)} \tag{5.2}$$

which is simply proportional to $g(H_b)$, i.e. the distribution of interaction fields in the Preisach-Néel model of hysteresis. This result can be generalized to real systems of interacting single-domain particles, in which case the vertical profile of the normalized FORC function

$$\rho^*(H_c,H_b)=\frac{\rho(H_c,H_b)}{\rho(H_c,0)} \tag{5.3}$$

yields the distribution of local interaction fields when taken at $H_c=0$ [*Egli*, 2006; *Li et al.*, 2012]. The width of this distribution is a function of the effective volume concentration of magnetic particles.

More generally, the vertical width of the FORC function, which is easily deduced from the normalized counterpart ρ^*, is a measure of magnetic memory loss as the applied field is increased from the value H_r where FORC measurements begin (the so-called reversal field) towards positive saturation. Perfect magnetic memory, such as that of non-interacting single-domain particles with squared hysteresis, implies that a measurement curve starting at a negative reversal field $H_r<0$ remains constant until the same field amplitude is reached in positive fields, i.e. $H=-H_r$. The resulting FORC diagram will then contain a sharp, horizontal ridge along $H_b=0$. Other contributions are added by reversible magnetic moment rotation in the applied field, yielding the characteristic FORC signature of non-interacting single-domain particles [*Newell*, 2005; *Egli et al.*, 2010].

As magnetic memory decreases, FORC magnetization begins to change before $H=-H_r$ is reached, giving a vertical spread to the central ridge, as seen in FORC diagrams of interacting single-domain particles [*Pike et al.*, 1999; *Egli*, 2006]. A similar phenomenon occurs with domain states: particles that are slightly too large for a single magnetic domain nucleate a single magnetic vortex as intermediate magnetization configuration between single-domain-like states, producing FORC. Nucleation and denucleation of such vortex states produce FORC diagrams with additional contributions below and above the central ridge [*Pike and Fernandez*, 1999; *Dumas et al.*, 2007; *Lappe et* *al.*, 2011]. The vertical spread of the FORC function increases with grain size, until a total loss of magnetic memory is reached with multidomain crystals, whose FORC diagram extends to very large H_b amplitudes [*Pike et al.*, 2001]. Because the reliability of absolute paleointensity records is related to the domain state of magnetic particles, and because this state is not always correctly deducible from bulk magnetic measurements (e.g. the Day plot), the vertical spread of FORC diagrams has been used as a selection tool for choosing most reliable rocks, avoiding time-consuming paleointensity experiments that will not pass quality checks [*Carvallo et al.*, 2006].

In order to address the abovementioned applications, `PlotFORC` offers the possibility to generate normalized diagrams where the vertical width of the FORC function is directly deducible from contour lines (Fig. 5.35a-b). This option is activated by setting

```
INPUT 21. Vertical profile normalization ......; Yes
```

in which case the normalized FORC diagram is plotted instead of the regular one. The default `INPUT 21` setting for the regular plot is `No`.

Because FORC diagrams are often not exactly symmetric about $H_b = 0$, normalization is performed with respect to the maximum amplitude of each vertical profile, i.e.

$$\rho^*(H_c, H_b) = \frac{\rho(H_c, H_b)}{\max\limits_{H_b}\left[\rho(H_c, H_b)\right]} \tag{5.4}$$

Usually, vertical profiles have a single maximum close to $H_b = 0$. Therefore, `PlotFORC` takes the first local maximum encountered from $H_b = 0$ upwards or downwards in each vertical profile as the value used for normalization. Because $\rho^* \le 1$, the distance of the contour defined by $\rho^* = 0.5$ from the central maximum (i.e., $\rho^* = 1$) coincides with the vertical half-width (HF) of the FORC function (Fig. 5.35b). Two contours, above and below the central maximum, can be used for this purpose; however, only the upper contour is recommended, because the lower quadrant of the FORC diagram is usually affected by reversible magnetic processes incompatible with the Preisach model (e.g., negative amplitudes in Fig. 5.35b). The example in Fig. 5.35 is based on interacting single-domain magnetite particles obtained from magnetotactic bacteria. Tear-drop contours (Fig. 5.35a) are characteristic for single-domain particles with moderate magnetostatic interactions [*Pike et al.*, 1999]. Contours of the normalized diagram (Fig. 5.35b) show maximum vertical spread at $H_c = 0$ and converge to a central ridge at larger coercivities, as calculated by *Egli* [2006]. The vertical half-width of the FORC function at $H_c = 0$ is ~13 mT: this is the typical strength of the local interaction field in this sample, from which a packing fraction of ~5% can be deduced for randomly oriented single-domain particles (see Fig. 8 in *Egli* [2006]).

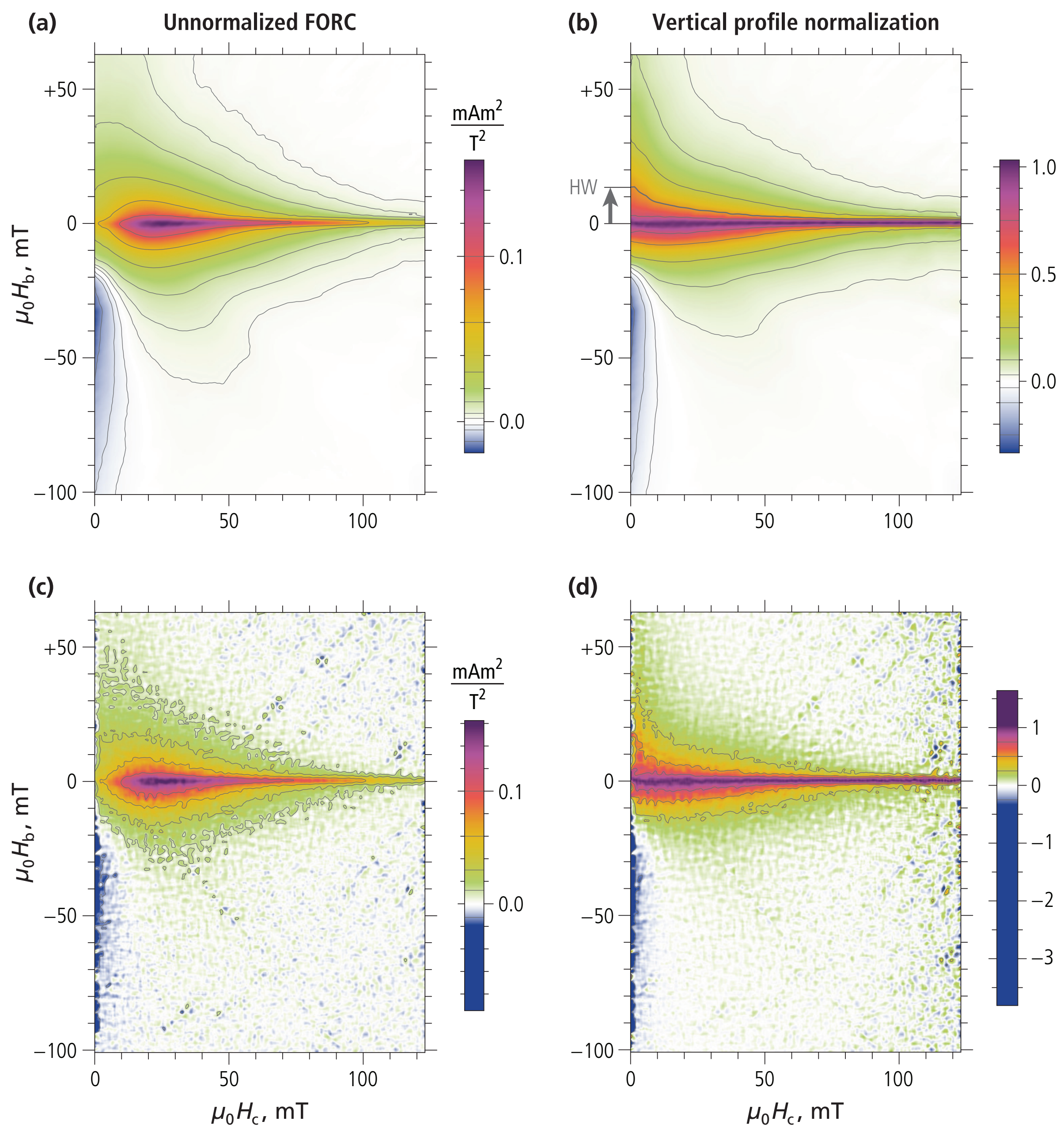


Fig. 5.35: Examples of vertical profile normalization (i.e. `INPUT 21` set to `Yes`) based on the FORC diagram of magnetosomes extracted from a magnetotactic bacteria culture (FORC data from *Wang et al.* [2013], reprocessed with VARIFORC). **(a)** Unnormalized FORC diagram generated with `CalculateFORC` and variable smoothing options. Contour lines have been added with `PlotFORC`. Teardrop-shaped contours and negative amplitudes in the lower quadrant are typical for moderately interacting single-domain particles. **(b)** Same as (a), after normalizing the FORC function with the maximum value in each vertical profile. The 0.5 contour line over the upper quadrant (highlighted) defines the half-width (HW) of vertical profiles. The half-width at $H_c = 0$ (arrow) coincides with the half-width of the interaction field distribution. **(c)** Unnormalized FORC diagram generated with a constant smoothing factor SF = 3. **(d)** Same as (c), after normalizing the FORC function with the maximum value in each vertical profile. Notice the measurement noise amplification near the right limit of the diagram, due to normalization with a small maximum amplitudes. Color scales in (a,c) and (b-d) are identical.

Measurement noise has detrimental effects on vertical profile normalization, especially over FORC regions with overall weak contributions (Fig. 5.35c-d). In this case, some maximum amplitudes found by `PlotFORC` might be controlled by local noise, rather than the characteristics of the FORC function. VARIFORC handles this problem at two levels. First, FORC diagram calculation with variable smoothing is very effective in eliminating local maxima generated by noise, as seen by comparing Fig. 5.35a and Fig. 5.35c. Second, local maxima are determined by polynomial regression over the same arrays of points used for smoothing. These maxima, if not significant, are discarded in favor of the FORC value at $H_b = 0$.

Finally, normalization by zero or negative values is avoided by taking half of the estimated standard error of the FORC function as the lowest maximum amplitude limit. As seen in Fig. 5.35d, measurement noise is amplified along vertical profiles with small maximum amplitudes, as for instance near $H_c = 0$ and near the right limit of the FORC diagram. If noise level becomes unacceptable, the plotted FORC diagram range can be reduced in order to exclude regions with insignificant contributions (see `INPUT 1` and `INPUT 2`).

The half-width of the FORC function is deduced from the contour lines corresponding to 0.5. More precisely, the half-width is given by the vertical distance between this contour line and the central maximum, which might be slightly shifted with respect to $H_b = 0$.

- Regions of the FORC diagram that are dominated by measurement noise or lack significant contributions are better excluded from the plotted range, since normalization yields meaningless results if profile maxima are not significant.

5.6. Literature

Carvallo, C., A.P. Roberts, R. Leonhardt, C. Laj, C. Kissel, M. Perrin, and P. Camps (2006). Increasing the efficiency of paleointensity analyses by selection of samples using first-order reversal curve diagrams, *Journal of Geophysical Research*, *111*, B12103, doi:10.1029/2005JB004126.

Dumas, R.K., C.P. Li, I.V. Roshchin, I.K. Schuller, and K. Liu (2007). Magnetic fingerprints of sub-100 nm Fe dots, *Physical Review B*, *75*, 134405.

Egli, R. (2006). Theoretical aspects of dipolar interactions and their appearance in first-order reversal curves of thermally activated single-domain particles, *Journal of Geophysical Research*, *111*, B12S17, doi:10.1029/2006JB004567.

Egli, R., A.P. Chen, M. Winklhofer, K.P. Kodama, and C.-S. Horng (2010). Detection of non-interacting single domain particles using first-order reversal curve diagrams, *Geochemistry*, *Geophysics*, *Geosystems*, *11*, Q01Z11, doi:10.1029/2009GC002916.

Egli, R. (2013). VARIFORC: an optimized protocol for the calculation of non-regular first-order curves (FORC) diagrams, *Global and Planetary Change*, *210*, 302-320, doi: 10.1016/j.gloplacha.2013.08. 003.

Heslop, D., and A.P. Roberts (2012). Estimation of significance levels and confidence intervals for first-order reversal curve distributions, *Geochemistry Geophysics Geosystems*, *13*, Q12Z40, doi: 10.1029/2012GC 004115.

Lappe, S.C., N.S. Church, T. Kasama, A. Bastos da Silva Fanta, G. Bromiley, R.E. Dunin-Borkowski, S. Russell, R.J. Harrison (2011). Mineral magnetism of dusty olivine: A credible recorder of pre-accretionary remanence, *Geochemistry Geophysics Geosystems*, *17*, Q12Z35, doi: 10.1029/2011GC 003811.

Li, J., W. Wu, Q. Liu, and Y. Pan (2012). Magnetic anisotropy, magnetostatic interactions, and identification of magnetofossils, *Geochemistry Geophysics Geosystems*, *13*, Q10Z51, doi: 10.1029/2012GC 004384.

Néel, L. (1958). Sur les effets d'un couplage entre grains ferromagnétiques, *Comptes Rendus de l'Académie des Sciences*, *246*, 2313-2319.

Newell, A.J. (2005). A high-precision model of first-order reversal curves (FORC) functions for single-domain ferromagnets with uniaxial anisotropy, *Geochemistry Geophysics Geosystems*, *6*, Q05010, doi:10.1029/2004 GC000877.

Pike, C., and A. Fernandez (1999). An investigation of magnetic reversal in submicron-scale Co dots using first order reversal curve diagrams, *Journal of Applied Physics*, *85*, 6668-6676.

Pike, C.R., A.P. Roberts, and K.L. Verosub (1999). Characterizing interactions in fine magnetic particle systems using first order reversal curves, *Journal of Applied Physics*, *85*, 6660-6667.

Pike, C.R., A.P. Roberts, M.J. Dekkers, and K.L. Verosub (2001b). An investigation of multi-domain hysteresis mechanism using FORC diagrams, *Physics of the Earth and Planetary Interiors*, *126*, 11-25.

Preisach, F. (1935). Über die magnetische Nachwirkung, *Zeitschrift für Physik*, *94*, 277-302.

Wang, H., D.V. Kent, and M.J. Jackson (2013). Evidence for abundant isolated magnetic nanoparticles at the Paleocene-Eocene boundary, *Proceedings of the National Academy of Sciences of the United States of America 110*, 425-430.