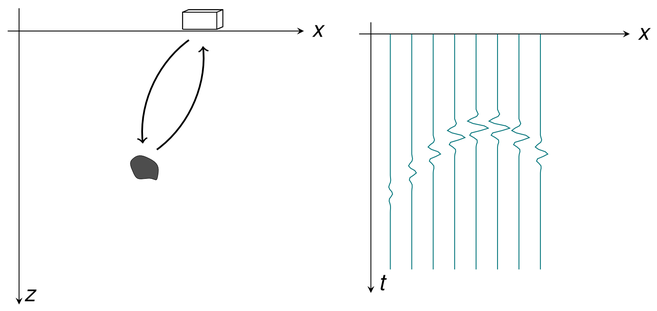
TECHNISCHE UNIVERSITÄT
ILMENAU

Optimizing Visualization and Documentation Process of Scientific Writing

Sayako Kodera

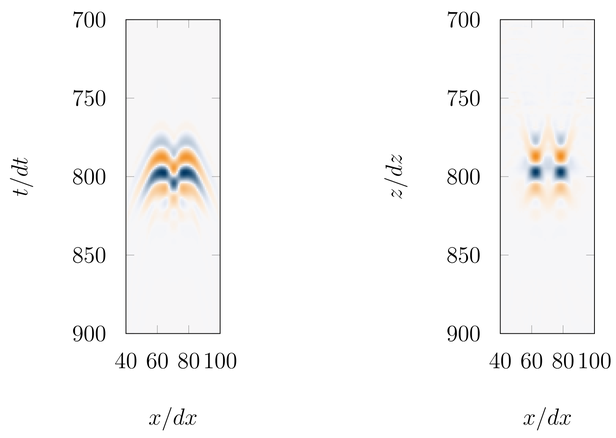
TU Ilmenau, Ultrasonic Imaging Team of EMS Research Group

Introduction



- Ultrasonic testing (UT) is one of the investigation methods in non-destructive testing (NDT)
- Discontinuities such as cracks or holes in test objects are localized
- Raw measurement data are often (post-)processed to enhance its imaging quality
- Data are conventionally visually analyzed

Motivations and Goals



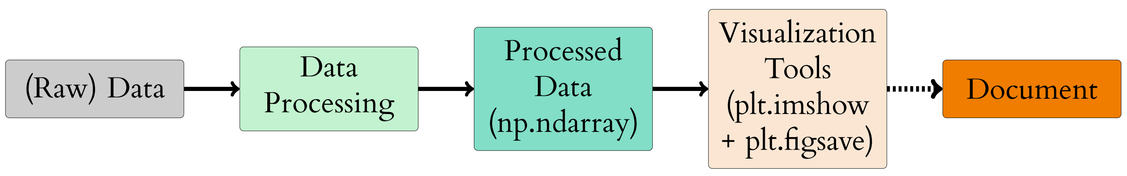
- Documents should have sufficient imaging quality for visual analysis
- The imaging quality should remain same even when it's enlarged
- Multiple images should be comparable in documents
- Changes in the data should be reflected in the document automatically
- Changes irrelevant to the data (color map, axis label etc) should be made fast and easily

Problems

Example approach : using plots directly in the document

- The imaging quality becomes poor, when the image is enlarged
- Keeping the coherency of the figure frames and styles is cumbersome
- The entire code should run again for making any changes

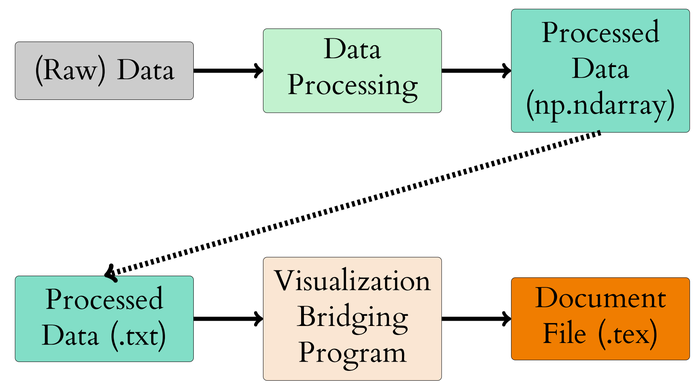
Workflow of the Example Approach



- The entire process is in a sequential chain
- However, only the processed data is required to be visualized
- The data processing part is most time consuming
- The documentation process is often not directly linked to the visualization tools

Solution

Separate the visualization process from data processing



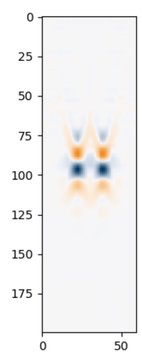
- Save the processed data at the end of the data processing chain
- Load the saved data in the visualization chain
- Convert the data into .tex-files with the bridging program
- Both chains can be done solely with Python
- Lastly insert the generated .tex file in the main document file

Bridging Program

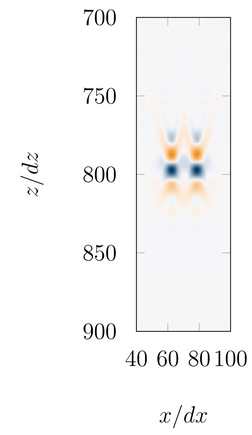
- Create a .tex script regarding pgfplot [1] [2]
- Specify axis details in the .tex script
- 1D case : use "\addplot coordinates" in the script and add x and y values from the loaded data
- 2D case : save the loaded data (np.ndarray) as png images with plt.imsave
- 2D case : use "\addplot graphics" in the script [2]
- Write a .tex-file with the generated script

Results

plt.figsave



Our Approach



- Figure frames are easily adjustable to the environment
- Details regarding axis labels and ticks can be modified directly in the document (without running other codes)

Summary

- Separate the visualization process from the data processing/generation
- Use visualization bridging programs to convert the data into .tex-files
- Changes irrelevant to the data can be made directly in the document
- **less stress with those "last minutes changes"**

References

[1] Pgfplots gallery. `http://pgfplots.sourceforge.net/gallery.html`. Online : last accessed in August 2018.

[2] Dr. Christian Feuersänger. Manual for package pgfplots. `http://pgfplots.sourceforge.net/pgfplots_layers.pdf`, 2012. Online : last accessed in August 2018.