

Fabian Brinkmann, Stefan Weinzierl

AKtools—an open software toolbox for signal acquisition, processing, and inspection in acoustics

Open Access via institutional repository of Technische Universität Berlin

Document type

Conference paper | Accepted version

(i. e. final author-created version that incorporates referee comments and is the version accepted for publication; also known as: Author's Accepted Manuscript (AAM), Final Draft, Postprint)

This version is available at

<https://doi.org/10.14279/depositonce-15094>

Citation details

Brinkmann, F., Weinzierl, S. (2017). AKtools—An Open Software Toolbox for Signal Acquisition, Processing, and Inspection in Acoustics. AES 142nd Convention, Berlin, Germany, 2017 May 20–23. (Engineering Brief 309, 2017 May) <http://www.aes.org/e-lib/browse.cfm?elib=18685>.

Terms of use

This work is protected by copyright and/or related rights. You are free to use this work in any way permitted by the copyright and related rights legislation that applies to your usage. For other uses, you must obtain permission from the rights-holder(s).

AKtools - an open software toolbox for signal acquisition, processing, and inspection in acoustics

Fabian Brinkmann and Stefan Weinzierl

Audio Communication Group, Technical University Berlin, Einsteinufer 17c, D-10587 Berlin, Germany

Correspondence should be addressed to Fabian Brinkmann (fabian.brinkmann@tu-berlin.de)

ABSTRACT

The acquisition, processing, and inspection of audio data plays a central role in the everyday practice of acousticians. However, these steps are commonly distributed among different and often closed software packages making it difficult to document this work. AKtools includes Matlab methods for audio playback and recording, as well as a versatile plotting tool for inspection of single/multi channel data acquired on spherical, and arbitrary spatial sampling grids. Functional blocks cover test signal generation (e.g. pulses, noise, and sweeps), spectral deconvolution, transfer function inversion using frequency dependent regularization, spherical harmonics transform and interpolation among others. Well documented demo scripts show the exemplary use of the main parts, with more detailed information in the description of each method. To foster reproducible research, AKtools is available under the open software European Union Public Licence (EUPL) allowing everyone to use, change, and redistribute it for any purpose: www.ak.tu-berlin.de/aktools.

1 Introduction

Reproducible research aims at providing all data that are necessary for completely repeating a study. Besides the research data itself – such as audio recordings, impulse responses, or ratings from listening tests – this especially includes the source code for acquiring and processing the data, and generating the graphics [1]. However, publishing all data is particularly challenging in empirical science where a large amount of data is often acquired, followed by a dedicated pre- and post-processing. This also implies that strategies for long term data availability (data savety) and traceability of changes (versioning) have to be considered in addition to mere publishing. In this context, AKtools are intended to foster reproducible research in fields that rely on the acquisition, processing, and inspec-

tion of acoustic signals. Possible strategies for providing the research data itself are given for example by DepositOnce [2] or the Open Repository for Research Data in Acoustics (OPERA - under construction) [3].

2 AKtools

2.1 Structure

AKtools is structured in three blocks that include (i) demo scripts with examples for functionality and function calls, (ii) the functions themselves, and (iii) demo data for illustrating certain applications. For ease of use, the installation of AKtools, as well as the functional scope and literature references are described in an accompanying read me file. Additionally, all function headers include a detailed list of input and

output parameters, and examples for function calls or a reference to a corresponding demo script. To ease the integration of AKtools into existing projects, the functionality is subdivided in small blocks that work on time signals in almost all cases. The time signals are organized in matrices of size $[N M C]$, where N is the number of samples, M the number of measurements, and C the number of channels. For efficiently switching between the time and frequency domain, AKtools contains functions for the conversion between single and both sided spectra in the case an even and uneven number of frequency bins, or samples, respectively.

2.2 Functional Scope

AKtools contains methods for the playback and recording, processing and inspection of acoustic signals, of which some are being introduced in the following. For a comprehensive overview, please refer to the demo scripts that are contained in AKtools, and which include use cases for the vast majority of the functional scope.

The sample precise single and multi channel audio playback and recording is realized by `AKio.m` which uses `playrec` [4], and `pa-wavplay` [5]. In combination with the sweep synthesis in the time or frequency domain (`AKsweepFD.m`, `AKsweepTD.m`), and the spectral deconvolution (`AKdeconv.m`), this furthermore enables versatile impulse response measurements, e.g. for the identification of non-linear distortion products using exponential sweeps [6], for the adaptive estimation of impulse responses using perfect sweeps [7], or measuring with constant signal-to-noise ratio across frequency using colored sweeps [8]. An example for measuring an impulse response and calibrating the frequency response of the measurement chain is given in `AKmeasureDemo.m`.

Besides the playback and recording of audio, the quick inspection and verification of signals by means of plotting plays a key roll. The function `AKp.m` holds a multitude of possibilities for this purpose, including simple two-dimensional time and frequency domain plots (Fig. 1a), and three-dimensional plots of data slices (Fig. 1b) or spherical data sets (Fig. 1c). Note that the Matlab code for generating all figures in this manuscript can be found in the Appendix.

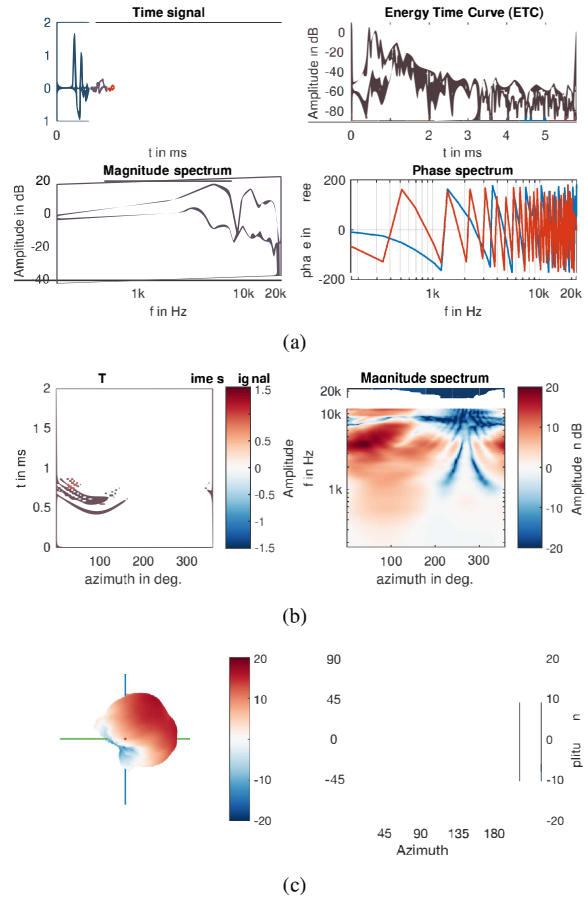


Fig. 1: Examples for graphics generated with `AKp.m` using head-related impulse responses (HRIRs) and transfer functions (HRTF) from the FABIAN data base [9]. (a) Left ear (blue) and right ear (red) HRIRs and HRTFs of a sound source to the left of the listener. (b) Left ear HRIRs and HRTFs magnitude in the horizontal plane, where 0° azimuth refers to a source in front of the listener, and 90° to a source to the left. (c) Spherical plots of the HRTFs magnitude at 6 kHz, where 90° elevation refers to a source above, and -90° to a source below the listener.

A simple yet often needed post-processing of acoustic signals is filtering. On this account, `AKfilter.m` provides numerous possibilities including high and low passes, band passes and rejections, cross-over networks, and octave filters, as well as high and low shelves, and parametric equalizers. Some examples are shown in

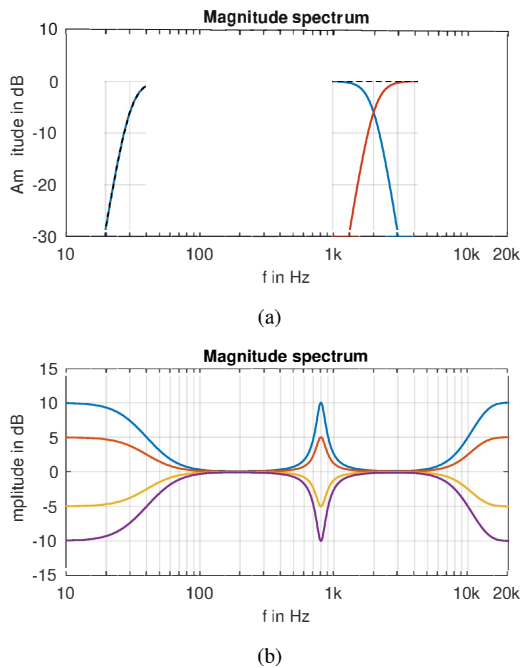


Fig. 2: Examples for filters generated with `AKfilter.m`: (a) Eighth order Linkwitz-Riley cross-over network (blue, red, $f_c = \{40, 2000\}$ Hz) and the addition of the channels (black). (b) Second order low-shelves and high-shelves ($f_c = \{40, 10.000\}$ Hz, Gain $\{-10, -5, 5, 10\}$ dB), and parametric equalizers with gains of $\{-10, -5, 5, 10\}$ dB ($f_c = 800$ Hz, $Q = 4$).

Fig. 2 – a more comprehensive overview is given in `AKfilterDemo.m`.

Another frequently required task is the transfer function inversion, be it for the equalization of headphones for binaural sound reproduction or the compensation of loudspeakers in room acoustics. `AKtools` holds the function `AKregulatedInversion.m` which implements a frequency dependent regularized LMS inversion that approaches a minimum phase or linear phase target function. By this means, the precision of the inversion can be restricted for certain frequency ranges, for example to avoid excessive gains in the inverse transfer function [10]. Fig. 3 shows the signals that are involved in the regularized inversion of a headphone transfer function (HpTF). In this case, the exact compensation of narrow but potentially inaudible notches at approximately 9 kHz and 13 kHz was avoided by

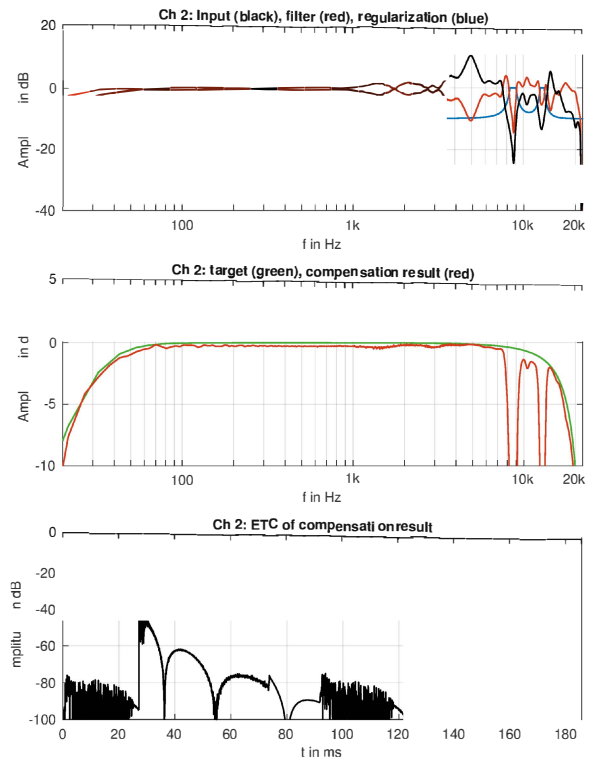


Fig. 3: Regulated inversion on the example of a headphone transfer function (HpTF). Top: HpTF (black), inverted HpTF (red) and regularization function (blue). Middle: target function (green) and inversion result (i.e. convolution of HpTF and inverted HpTF, red). Bottom: Energy time curve ($x[n]^2$) approaching a delayed minimum phase system.

means of a regularization function composed from two parametric equalizers.

`AKtools` moreover includes basic tools for spherical harmonics processing including the discrete spherical harmonics transform (DSHT) that decomposes spherical data sets (e.g. head-related transfer functions – HRTFs) into a weighted superposition of orthogonal spherical basis function shown in Fig. 4(a). Besides a significant reduction in data this comes with the advantage of a spatially continuous data representation. The DSHT and its inverse according to eq. (3.34) in Rafaely (2015) [11] are implemented in `AKsht.m`, and `AKisht.m` – they can be applied either to complex spectra or separately on magnitude and un-

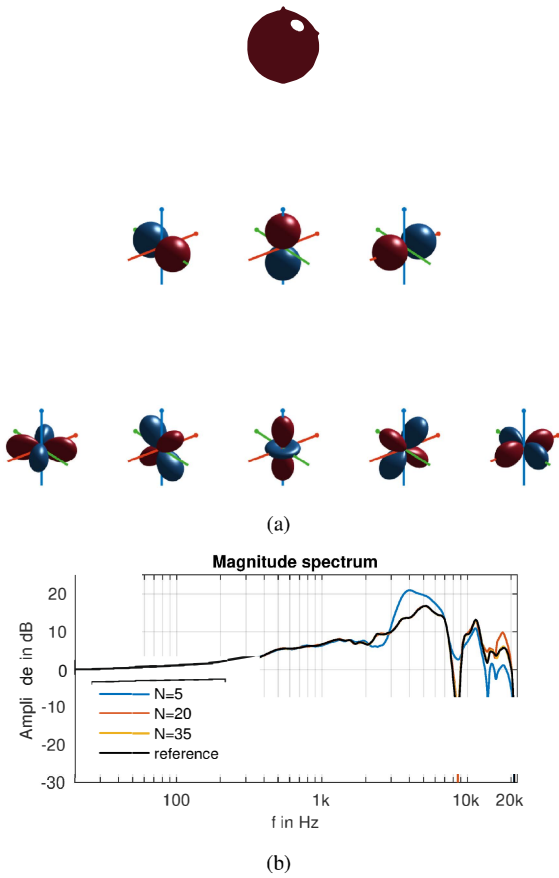


Fig. 4: (a) Spherical harmonics basis function according [12, 11], and for orders 0 (top row), 1 (middle row) and 2 (bottom row). For ease of display, the two leftmost columns show the imaginary part of the complex functions and the 3 rightmost rows the real part. The colors denote positive (red), and negative (blue) values. Red, blue, and green lines point in positive x, y, and z direction, respectively. (b) Head-related transfer function for the left ear of FABIAN [9] and a source to the left before (black), and after discrete spherical harmonics transform with different truncation orders N .

wrapped phase values. Examples for an HRTF before and after spherical harmonics transform are depicted in Fig. 4(b) for different truncation orders. The DSHT requires full spherical data sets, a prerequisite that is often violated by acoustically measured data. A possible solution to this problem is implemented

in `AKsphericalCapExtrapolationDemo.m`. Following Ahrens *et al.* (2012) [13], a low order DSHT ($3 \lesssim N \lesssim 4$) is performed on the incomplete data to complete the spatial sampling grid, and a high order DSHT is applied to the completed data in a second step.

2.3 Head-related transfer functions

AKtools are accompanied by the publicly available FABIAN head-related transfer functions data base [9] that holds acoustically measured and numerically simulated HRTFs for eleven head-above torso orientations, and a full-spherical, high resolution sampling grid. For convenience, HRTFs are also provided by means of high order ($N=35$) spherical harmonics coefficients. HRTFs interpolated to arbitrary source positions and head-above-torso orientations can be obtained using `AKhrirInterpolation.m` which interpolates the source positions in the spherical harmonics domain and the head-above-torso orientation separately on the magnitude and unwrapped phase spectra following Brinkmann *et al.* (2015) [14].

2.4 Availability

The AKtools are available under the European Union Public Licence (EURL) granting unrestricted access, and can be obtained from www.ak.tu-berlin/aktools. Versioning of the source code is done by subversion (SVN), and the backup system for data safety is maintained by the Technical University Berlin. AKtools need Matlab 2013 or later including the Signal Processing, and Statistics and Machine Learning Toolbox. However, the majority of functionality is also available without any additional toolboxes. Some demo scripts within AKtools make use of the FABIAN head-related transfer function data base [9] which is available from <http://dx.doi.org/10.14279/depositonce-5718>.

3 Summary

The free Matlab toolbox AKtools was introduced which can be used for common tasks in acoustics such as audio playback and recording, impulse response measurements, filtering, transfer function inversion, or HRTF interpolation. AKtools is an ongoing project, that will be extended in the future and welcomes third party contributions.

Acknowledgement

AKtools has contributions from: Alexander Lindau, Andreas Rotter, André Giese, David Ackermann, Daniel Kahlhöfer, Dmitry Grigoriev, Fabian Brinkmann, Frank Schultz, Hannes Helmholz, Marc Voigt, Omid Kokabi, Zora Schärer Kalkandjiev. AKtools uses third party source code from: Matt Frear and Josph Desloge (pa_wavplay), Robert Humphrey (playrec), Piotr Majdak *et al.* (SOFA Matlab API), Malcolm Slaney (auditory toolbox), Carlos Adrian and Vargas Aguilera (cbfreeze, cbhandle), Doug Hall (makeColormap), Duane Hanselman (mmpolar), Felipe G. Nievinski and Pekka Kumpulainen (subtightplot), John Burkhardt (j_polynomial), color specifications and designs by Cynthia Brewer (colorbrewer.org). AKtools was inspired by the ITA-Toolbox [15].

References

- [1] Vandewalle, P., Kovačević, J., and Vetterli, M., “Reproducible research in signal processing – what, why, and how,” *IEEE Signal Processing Magazine*, 26(3), pp. 37–46, 2009.
- [2] Technical University Berlin, “Deposit Once – Repository for research data and publications,” 2016.
- [3] Weinzierl, S., “An Open Repository for Research Data in Acoustics (OPERA),” in *Fortschritte der Akustik – DAGA 2017*, Kiel, Germany, 2017.
- [4] Humphrey, R., “Playrec – Multichannel Matlab audio,” <http://www.playrec.co.uk>, 2008.
- [5] Desloge, J. G., “pa-wavplay for 32-bit and 64-bit,” <https://de.mathworks.com/matlabcentral/fileexchange/47336-pa-wavplay-for-32-bit-and-64-bit>, 2014.
- [6] Novák, A., Simon, L., Kadlec, F., and Lotton, P., “Nonlinear system identification using exponential swept-sine signal,” *IEEE Trans. on Instrumentation and Measurement*, 59(8), pp. 2220–2229, 2010.
- [7] Antweiler, C., Telle, A., Vary, P., and Enzner, G., “Perfect-sweep NLMS for time-variant acoustic system identification,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011.
- [8] Müller, S. and Massarani, P., “Transfer function measurement with Sweeps. Directors cut including previously unreleased material and some corrections,” *J. Audio Eng. Soc.* (Original release), 49(6), pp. 443–471, 2001.
- [9] Brinkmann, F., Lindau, A., Weinzierl, S., Geissler, G., van de Par, S., Müller-Trapet, M., Opdam, R., and Vorländer, M., “The FABIAN head-related transfer function data base,” <http://dx.doi.org/10.14279/depositonce-5718>, 2016.
- [10] Norcross, S. G., Bouchard, M., and Soulodre, G. A., “Inverse Filtering design using a minimal phase target function from regularization,” in *121th AES Convention, Convention Paper 6929*, San Francisco, USA, 2006.
- [11] Rafaely, B., *Fundamentals of spherical array processing*, Springer, Berlin, Heidelberg, Germany, 1st edition, 2015, doi:<https://doi.org/10.1007/978-3-662-45664-4>.
- [12] Williams, E. G., *Fourier Acoustics. Sound radiation and nearfield acoustical holography*, Academic Press, 1 edition, 1999.
- [13] Ahrens, J., Thomas, M. R. P., and Tashev, I. J., “HRTF magnitude modeling using a non-regularized least-squares fit of spherical harmonics coefficients on incomplete data,” in *AP-SIPA Annual Summit and Conference*, Hollywood, USA, 2012.
- [14] Brinkmann, F., Roden, R., Lindau, A., and Weinzierl, S., “Audibility and interpolation of head-above-torso orientation in binaural technology,” *IEEE J. Sel. Topics Signal Process.*, 9(5), pp. 931–942, 2015, doi:<https://doi.org/10.1109/jstsp.2015.2414905>.
- [15] Dietrich, P., Guski, M., Pollow, M., Müller-Trapet, M., Masiero, B., Scharrer, R., and Vorländer, M., “ITA-Toolbox – An open source Matlab toolbox for acousticians,” in *Fortschritte der Akustik – DAGA 2012*, pp. 151–152, Darmstadt, Germany, 2012.

Appendix

This appendix demonstrates some AKtools basics and generates all figures that are shown in this manuscript. To start using AKtools download it from www.ak.tu-berlin.de/aktools navigate to the AKtools folder in Matlab and run AKtoolsStart.m. This source code was tested with AKtools revision 33, and Matlab 2016a.

```
%% ----- PLOTTING
% get HRIRs on an evenly sampled grid
g = AKgreatCircleGrid(-90:2:90, 2, 90);
[l, r] = AKhrirInterpolation(g(:,1), g(:,2), 0, ...
    'measured_ir');

%% get and plot HRIR to the left of the listener
id = AKsubGrid(g, 'any', [90 0]);

AKf(18,9)
AKp(l(:,id) r(:,id))
print('-dpdf', 'figure1a')

%% get and plot the horizontal plane
id = AKsubGrid(g, 'transverse', 0);

AKf(18,6)
subplot(1,2,1)
AKp(l(:,id), 't3d', 'y', g(id,1), 'dr', ...
    [-1.5 1.5], 'x', [0 2])
xlabel 'azimuth in deg.'
subplot(1,2,2)
AKp(l(:,id), 'm3d', 'y', g(id,1), 'dr', ...
    [-20 20])
xlabel 'azimuth in deg.'

print('-dpdf', 'figure1b')

%% plot spherical HRTF dataset at 6 kHz
AKf(18,6)
subplot(1,2,1)
AKp(l, 's2', 'az', g(:,1), 'el', g(:,2), ...
    'dr', [-20 20], 'sph_f', 6000, ...
    'labeling', 'off', 'hp_view', [90 0])
subplot(1,2,2)
AKp(l, 's5', 'az', g(:,1), 'el', g(:,2), ...
    'dr', [-20 20], 'sph_f', 6000)
set(gca, 'xTick', 1:90:361)

print('-djpeg', '-r300', 'figure1c')
```

```
%% ----- FILTERING
% generate dirac signals for filtering
x = AKdirac(2^14, 1);

%% apply high-pass, and cross-over
y = AKfilter(x, 'hp', 30, 0, 44100, 8, 'LR');
y = AKfilter(y, 'xover', 2000, 0, 44100, 8, 'LR');

AKf(12,6)
AKp(squeeze(y), 'm2d', 'x', [10 20000])
AKp(sum(y,3), 'm2d', 'x', [10 20000], 'dr', ...
    [10 -30], 'dash', 1)
print('-dpdf', 'figure2a')
```

```
%% apply low/high-shelves, and parametric EQs
y = AKfilter(x, 'ls', 40, [10 5 -5 -10], ...
    44100, 2, 'mid');
y = AKfilter(y, 'hs', 10000, [10 5 -5 -10], ...
    44100, 2, 'mid');
y = AKfilter(y, 'peq', 800, [10 5 -5 -10], ...
    44100, 4, 'hpl', 'cos');

AKf(12,6)
AKp(y, 'm2d', 'x', [10 20000], 'dr', [-15 15])
print('-dpdf', 'figure2b')
```

```
%% ----- SPHERICAL HARMONICS BASIS FUNCTIONS
% get spherical harmonics up to order n
N = 2;
[Ynm, n, m] = AKsh(N, [], g(:,1), 90-g(:,2));

AKf(10)
% plot imaginary part for m<0, and real part ...
for m>=0
for k = 1:numel(n)
if m(k)<0;
SH = imag(Ynm(:,k));
else
SH = real(Ynm(:,k));
end

subtightplot(N+1,2*N+1, n(k)*(2*N+1) + N+1 ...
    + m(k), 0)
AKp(SH, 'x7', 'az', g(:,1), 'el', g(:,2), ...
    'cb', 0, 'sph_proc', 'tri')
title('')
end

print('-dpng', '-r300', 'figure4a')
```

```
%% ----- SPHERICAL HARMONICS TRANSFORM
% spherical harmonics transform
% (left ear HRIRs, SH order N=35)
f_nm = AKsht(1, true, [g(:,1) 90-g(:,2)], 35, ...
    'complex');

%% inverse transform, different truncation orders
% (source on horizontal plane, 90 deg. to the left)
N = [5 20 35];
for nn = 1:numel(N);
l_sh(:,nn) = AKisht(f_nm(1:(N(nn)+1)^2,:), ...
    true, [90 90], 'complex');
end

%% plot reference and order truncated HRTFs
id = AKsubGrid(g, 'any', [90 0]);

AKf(12,6)
AKp(l_sh, 'm2d', 'N', 4410)
AKp(l(:,id), 'm2d', 'N', 4410, 'dr', [-30 25])
legend('N=5', 'N=20', 'N=35', 'reference', ...
    'location', 'SouthWest')
print('-dpdf', 'figure4b')
```

```
%% ----- REGULATED INVERSION
% run demo script - figure 3 is saved automatically
AKregulatedInversionDemo
```