Representative-volume sizing in finite cylindrical computed tomography by low-wavenumber spectral convergence

Supplementary Material: Software Documentation

Fernando Alonso-Marroquin, Abdullah Alqubalee, and Christian Tantardini

January 13, 2026

Software purpose and scope

The MATLAB software is intended for the analysis of long cylindrical core samples (e.g., μCT volumes or other 3D imaging modalities) in which spatial properties such as porosity, phase fraction, or burrow-related indicators may exhibit slow axial variation. In sufficiently long specimens, these low-frequency changes can act as a non-stationary trend along the core axis and can bias statistical descriptors that implicitly assume stationarity. The software provides a reproducible workflow to (i) preprocess cylindrical 3D datasets (binarization and enforcement of cylindrical support), (ii) separate slow axial trends from residual fluctuations using rolling-window detrending, and (iii) quantify whether the residual field is compatible with an approximately stationary description over the scales of interest.

Two complementary analyses are implemented. First, an axial analysis reduces the 3D volume to a one-dimensional slice-wise signal (e.g., mean phase fraction per slice) and applies a moving-average detrending over a range of window sizes. Candidate windows are evaluated using the excess kurtosis of the detrended residual as a compact indicator of Gaussian-likeness, supporting a practical choice of detrending scale. Second, a transverse (slice-wise) spatial analysis evaluates representativeness within a single cross-sectional slice by computing disk-averaged estimates of the target property as the sampling radius increases, together with second-order descriptors (two-point covariance and the corresponding radial spectrum) computed within an inscribed circular region.

A MATLAB scripts: algorithms, workflow, and dependencies

Function-designation convention. Throughout this appendix, each function call is labeled as one of:

- **native:** available in base MATLAB (no specialized toolbox required);
- **native_IPT:** available in MATLAB's Image Processing Toolbox (IPT);
- **project:** custom function distributed with this project.

A.1 `playground_detrending_field.m`

Purpose. Perform an axial stationarity-oriented diagnostic on a 3D cylindrical volume by: (i) binarizing and masking the volume, (ii) computing a slice-wise phase fraction signal $\phi(k)$ along the z direction, (iii) detrending $\phi(k)$ using moving-average windows of varying size, and (iv) evaluating the detrended residual via its excess kurtosis.

Algorithm 1 Axial detrending and excess-kurtosis sweep on a cylindrical 3D volume

Require: 3D TIFF volume file `rockname.tif`; window range $\mathcal{W} = \{3, \dots, 100\}$
Ensure: Slice-wise phase fraction $\phi(k)$; excess kurtosis curve $K_{\text{ex}}(w)$; exported PDF figures

Read volume: $A \leftarrow$ `tiffreadVolume(...)` *(native_IPT)*
Normalize intensities: $A_d \leftarrow$ `mat2gray(A)` *(native_IPT)*
Compute global threshold: $T \leftarrow$ `graythresh(`$A_d(:)$`)` *(native_IPT)*
Binarize: $B \leftarrow (A_d > T)$ *(native)*
Mask cylinder: $B \leftarrow$ `cylinder_masked(B)` *(project)*
Store voxel geometry: $(N_x, N_y, N_z) \leftarrow$ `size(B)` *(native)*
Compute slice-wise phase fraction:
for $k \leftarrow 1$ **to** N_z **do**
    $x(k) \leftarrow k$ *(native)*
    $\phi(k) \leftarrow \text{mean}\big(B(:,:,k)(:)\big)$ *(native)*
end for
Sweep rolling-window detrending:
for all $w \in \mathcal{W}$ **do**
    $K_{\text{ex}}(w) \leftarrow$ `detrend_window_xy(x,`ϕ`,w)` *(project)*
end for
Export diagnostic curve: plot $K_{\text{ex}}(w)$ vs w and save as `kurtosis.pdf` *(native)*
Select window: choose w_{opt} (criterion-dependent) *(native)*
Export detrending visualization: run `detrend_window_xy(x,`ϕ`,` w_{opt}`)` and save as `MA.pdf` *(project + native)*

Graphical outputs.

- `kurtosis.pdf`: excess kurtosis vs moving-average window size (w vs K_{ex}).
- `MA.pdf`: detrending visualization produced by `detrend_window_xy` at the selected window w_{opt}.

Required project functions.

- `cylinder_masked.m`
- `detrend_window_xy.m`

A.2 `playground_optimal_REV_radius.m`

Purpose. Compute slice-wise spatial diagnostics from a single 2D cross-sectional slice extracted from a 3D cylindrical volume. The script computes: (i) the disk-averaged phase fraction $\langle\phi\rangle(r)$, (ii) the covariance $C(r)$ within an inscribed circular region, (iii) the corresponding radial spectrum $\widehat{C}(k_r)$, and (iv) a single 2×2 figure summarizing the results.

Algorithm 2 Slice-wise radial averaging, covariance, and spectral diagnostics on a μCT volume

Require: 3D TIFF file `filename`; slice index z_0; radius list `rList`; parameters `doCircleCrop`, `fracR`, `lagStep`, `Nk`

Ensure: Disk-average curve $\langle\phi\rangle(r)$; covariance $C(r)$; spectrum $\widehat{C}(k_r)$; exported figure `REV_r.pdf`

Read volume: $V \leftarrow$ `tiffreadVolume(filename)` *(native_IPT)*

Binarize and mask:

  $V_n \leftarrow$ `mat2gray(V)` *(native_IPT)*

  $B \leftarrow$ `imbinarize(`V_n`)` *(native_IPT)*

  $B \leftarrow$ `cylinder_masked(B)` *(project)*

Extract slice and set geometry:

  $B_{\text{slice}} \leftarrow B(:,:,z_0)$ *(native)*

  $(N_x, N_y) \leftarrow$ `size(`B_{slice}`)`; $R_{\max} \leftarrow \lfloor \min(N_x, N_y)/2 \rfloor$ *(native)*

Compute disk-averaged phase fraction:

for all $r \in$ `rList` **do**

  $\langle\phi\rangle(r) \leftarrow$ `circleMeanPlot(`B_{slice}`, r, false)` *(project)*

end for

Compute 2D covariance:

  $(rvals, C_r) \leftarrow$ `slice2cov_bool(`B_{slice}`, doCircleCrop, fracR, lagStep)` *(project)*

Compute radial spectrum:

  $(k_r, \widehat{C}) \leftarrow$ `radialSpectrum2D(rvals, `C_r`, Nk)` *(project)*

Assemble 2×2 tiled figure:

  Use `tiledlayout/nexttile` to plot (a) slice with overlay circles, (b) $\langle\phi\rangle(r)$, (c) $C(r)$, (d) $\widehat{C}(k_r)$ *(native)*

Export: `print(fig,'REV_r.pdf','-dpdf','-painters')` *(native)*

Graphical output.

- `REV_r.pdf`: 2×2 tiled figure containing (a) the binarized slice with overlay circles at $r = r_{\text{overlay}}$ and $r = R_{\max}$, (b) $\langle\phi\rangle(r)$ vs radius, (c) covariance $C(r)$ vs lag distance, and (d) radial spectrum $\widehat{C}(k_r)$ vs k_r (log–log).

Required project functions.

- `cylinder_masked.m`
- `circleMeanPlot.m`
- `slice2cov_bool.m`
- `radialSpectrum2D.m`