==

FRACTIONAL TEMPORAL SEIR MODEL WITH SIGNORINI BOUNDARY CONDITIONS

Authors: Omar Elamraoui, EL Hassan Essoufi, Abderrahim Zafrar

This supplementary material contains MATLAB code for simulating, analyzing, and visualizing a fractional-order SEIR epidemic model with memory effects and Signorini boundary conditions, as presented in:

"Switching Cases for Fractional Time SEIR Model with Memory and Space Diffusion"

==

FILES INCLUDED

📂 Main Simulation and Analysis Files

🔹 `Main_SEIR_Frac_Memory.m`
Main driver script for the SEIR model with fractional temporal memory and Signorini boundary conditions.

- Runs simulations and generates output plots.
- Configurable for different parameter studies (e.g., infection rate, recovery rate).

🔹 `show_Sensitivity_Param.m`
Driver script to:

- Perform sensitivity analysis by varying a parameter.
- Generate plots for S, E, I, R over time for each parameter setting.

🔹 `SEIR_Show_var_alpha.m`
Script for:

- Studying and plotting the effect of varying the fractional-order parameter `alpha` on the SEIR dynamics.

🔹 `SEIR_var_alpha_fun.m`
Function supporting `SEIR_Show_var_alpha.m`:

- Runs simulations across different `alpha` values.
- Returns the time evolution of SEIR components.

📂 Core Functions

🔹 `analyze_param_impact.m`

- Runs SEIR simulations for a range of values of a specified parameter (e.g., `\beta`, `\gamma`, `\alpha`).
- Returns S, E, I, R solutions in cell arrays for easy comparison.

🔹 `run_SEIR_simulation.m`

- Solves the fractional-order SEIR model over space and time using FEM and Uzawa's method.
- Incorporates memory effects and Signorini boundary conditions.
- Returns S, E, I, R matrices over time.

🔹 `UzawaSignoriniSolver.m`

- Solves the variational inequality for Signorini boundary conditions using a regularized Uzawa method.
- Returns solution vector, boundary projections, Lagrange multipliers, and error metrics.

📂 Helper Functions

🔹 `kpde2dumsh.m`, `kpde2dstf.m`, `kpde2drhs.m`, `kpde2dmss.m`

- Generate meshes, assemble finite element matrices, and compute RHS vectors.

==

REQUIREMENTS

✅ MATLAB R2019b or newer (recommended)
✅ No specialized toolboxes required — only standard MATLAB functions

==

ACKNOWLEDGMENTS

Finite element matrix assembly methods adapted from: Koko J., *Fast finite element assembly in MATLAB and Octave*, 2016.
https://perso.isima.fr/~jokoko/codes.html

==

CONTACT

For questions or further details, please contact:
Omar Elamraoui (oelamraoui34@gmail.com)