

# JDD Method Code

Contact: Rebecca Menssen  
[beccagobel@gmail.com](mailto:beccagobel@gmail.com)

Intro:

What you will find in this repository are example codes for implementing the JDD code in one, two, and three dimensions. These examples walk through how to do JDD analysis, and can be edited for use with experimental data. Questions, comments, or alerts to any bugs should be directed to me, and I will do my best to answer in a timely method!

These codes are meant to serve as examples. They can undoubtedly be improved through improved parameter fitting, MSD seeding, integration, etc. In many cases though, they should work well from the start, and then can be edited to suit individual needs. They should be reasonably well documented, but if something is unclear, feel free to reach out, or its likely that the corresponding code for a different dimension might have better commenting.

We hope this tool can be of use to many researchers, and by removing barriers like figuring out how to code it up, that it can be more accessible.

- **1D:** Codes relevant to 1D JDD method
  - **Experiments:** Functions that walk through how to use the JDD method in 1D for non-correlated data (i.e. JDD not assembled with sliding).
    - *ExperimentD1D.m*: Pure Diffusion example code
    - *ExperimentV1D.m*: Directed Diffusion example code
    - *ExperimentA1D.m*: Anomalous Diffusion example code
    - *ExperimentDD1D.m*: Double Diffusion combination model example code
    - *ExperimentDV1D.m*: Pure-Directed combination model example code
    - *ExperimentDA1D.m*: Pure-Anomalous combination model example code
  - **Other Functions:** Functions for jump distance, model fitting, and integration
    - *JumpDistance1D.m*: function to calculate non-sliding jump distance when trajectories are length time lag +1
    - *jddswEEP.m*: Function to calculate sliding jump distance with a set time lag.
    - *ModelFitting1D.m*: Function to perform parameter fitting on D,V, and A models with MSD seeding.
    - *ModelFitting1DwithSeeds.m*: Function to perform parameter fitting on D,V, and A models with user defined seeds
    - *ModelFitting1DwithComboModels.m*: Function to perform parameter fitting on D, V, A, DD, DV, and DA models with msd seeding.
    - *Integration1D.m*: Function for model selection for D,V, and A models.
    - *Integration1DwithComboModels.m*: Function for model selection for D, V, A, DD, DV, and DA models.
    - *intfuncD.m*: Function used within integration scheme for D model.
    - *intfuncV.m*: Function used within integration scheme for V model.

- *intfuncA.m*: Function used within integration scheme for A model.
  - *intfuncDD.m*: Function used within integration scheme for DD model.
  - *intfuncDV.m*: Function used within integration scheme for DV model.
  - *intfuncDA.m*: Function used within integration scheme for DA model.
- Trajectory simulators: Functions to simulate trajectories
  - *Diffusion1D.m*: trajectory simulator for pure diffusion.
  - *DirectedDiffusion1D.m*: trajectory simulator for directed diffusion.
  - *AnomalousDiffusion1D.m*: trajectory simulator for anomalous subdiffusion.
- **2D**: Codes relevant to 2D JDD method
  - **Experiments**: Functions that walk through how to use the JDD method in 1D for non-correlated data (i.e. JDD not assembled with sliding).
    - *ExperimentD2D.m*: Pure Diffusion example code
    - *ExperimentV2D.m*: Directed Diffusion example code
    - *ExperimentA2D.m*: Anomalous Diffusion example code
  - **Other Functions**: Functions for jump distance, model fitting, and integration
    - *JumpDistance2D.m*: function to calculate non-sliding jump distance when trajectories are length time lag +1
    - *jddswEEP2D.m*: Function to calculate sliding jump distance with a set time lag.
    - *jddswEEP2Dintervals.m*: Function to calculate sliding jump distance with a set time lag, but to keep intervals independent.
    - *ModelFitting2D.m*: Function to perform parameter fitting on D,V, and A models with MSD seeding.
    - *Integration2D.m*: Function for model selection for D,V, and A models.
    - *intfuncD2D.m*: Function used within integration scheme for D model.
    - *intfuncV2D.m*: Function used within integration scheme for V model.
    - *intfuncA2D.m*: Function used within integration scheme for A model.
  - **Functions for Combination Models**: Functions that can be used to do the JDD method with combination models
    - *ModelFitting2DexperimentaldataCombo*: Example code for model fitting in 2D with combination models and utilizing the MSD
    - *ModelFitting2DexperimentaldataCombOWithSeeds*: Example code for model fitting in 2D with combination models and utilizing the MSD
    - *Integration2DwithComboModels.m*: Function for model selection for D, V, A, DD, DV, DA, and VA models.
    - *intfuncDD.m*: Function used within integration scheme for DD model.
    - *intfuncDV.m*: Function used within integration scheme for DV model.
    - *intfuncDA.m*: Function used within integration scheme for DA model.
    - *intfuncVA.m*: Function used within integration scheme for VA model.
  - **Trajectory simulators**: Functions to simulate trajectories
    - *Diffusion2D.m*: trajectory simulator for pure diffusion.
    - *DirectedDiffusion2D.m*: trajectory simulator for directed diffusion.
    - *AnomalousDiffusion2D.m*: trajectory simulator for anomalous subdiffusion.

- **3D:** Codes relevant to 3D JDD method
  - **Experiments:** Functions that walk through how to use the JDD method in 1D for non-correlated data (i.e. JDD not assembled with sliding).
    - *ExperimentD3D.m*: Pure Diffusion example code
    - *ExperimentV3D.m*: Directed Diffusion example code
    - *ExperimentA3D.m*: Anomalous Diffusion example code
  - **Other Functions:** Functions for jump distance, model fitting, and integration
    - *JumpDistance3D.m*: function to calculate non-sliding jump distance when trajectories are length time lag +1
    - *jddsweep3D.m*: Function to calculate sliding jump distance with a set time lag.
    - *ModelFitting3D.m*: Function to perform parameter fitting on D,V, and A models with MSD seeding.
    - *Integration3D.m*: Function for model selection for D,V, and A models.
    - *intfuncD.m*: Function used within integration scheme for D model.
    - *intfuncV.m*: Function used within integration scheme for V model.
    - *intfuncA.m*: Function used within integration scheme for A model.
  - **Trajectory simulators:** Functions to simulate trajectories
    - *Diffusion3D.m*: trajectory simulator for pure diffusion.
    - *DirectedDiffusion3D.m*: trajectory simulator for directed diffusion.
    - *AnomalousDiffusion3D.m*: trajectory simulator for anomalous subdiffusion.
- **Common Functions**
  - Contains a few functions to help with making the JDD like binning, and also for MSD seeding (polyfit zero).