Model Name:

Jordan Water Model (JWM), incl. the extension for Tanker Water Market Analyses (JWM-T), version 2.0

New in v2.0: Selection of pre-parameterized JWM and JWM-T simulations via “model_setup.xlsx”

Model Contact:

Jim Yoon (jimyoon@gmail.com, JWM), Christian Klassert (christian.klassert@ufz.de, JWM-T)

Software Requirements:

The model has been developed and tested on a PC machine in Windows 10. A future model run (years 2016-2100) requires approximately 4GB of RAM.

The JWM has been developed and tested in Python 2.7.16. We recommend using the Anaconda distribution of Python, as the model relies on several scientific packages pre-installed with Anaconda. The following packages are required for the model to run (versions that were tested have also been specified where applicable):

- pynsim (0.1.4)
- numpy (1.11.3)
- pandas (0.22.0)
- pyomo (5.6.9)
- scipy (0.17.0)
- pickle
- os
- time
- zipfile
- logging
- shapefile
- math
- datetime
- itertools
- calendar
- ast

Compatibility issues have been encountered with certain combinations of numpy, pandas, pyshp, xlrd, and xlsxwriter packages, as well as between pyomo and pyutilib. The following versions are confirmed to work for the JWM:

- numpy 1.11.3
- pandas 0.22.0
- pyshp 1.2.10
- xlrd 0.9.4
- xlsxwriter 0.8.4

- pyutilib 5.8.0
- pyomo 5.6.9

Additionally, the model relies on the following software packages for various optimization routines, which are either open-source or available via free academic licenses.

- CPLEX (version 12.8.0)
- GLPK (version 4.65)
- IPOPT (version 3.11.1)

Installation Tips:

- Add the directory that the "JWM" folder resides in to your system's pythonpath variable.
- For the system to appropriately identify external solver executables during run time, paths to the appropriate solvers (associated with the optimization solvers listed above) should be added to your system path variable. These executables include:
    1. cplex.exe
    2. glpsol.exe
    3. ipopt.exe

Pynsim:

The JWM utilizes the Python Network Simulator package (pynsim) for core simulation functionality (version 0.1.4). Full documentation on pynsim can be found at the following github repository: https://github.com/UMWRG/pynsim

Running the Model and Processing Results:

The model can be run by executing the main.py file. For every component in the network, the model saves the history of attributes defined in the component's properties dictionary (see pynsim documentation for further details). To interrogate model results, script can be added to the bottom of the main.py file to extract history for component properties of interest. Alternatively, the code from main.py can be processed through an interactive Python console to interrogate model results interactively. An example script for extracting model output which is utilized for processing simulations and outputting results to csv for further post-processing is provided as "example_output_extraction.py."

Model Options:

Pre-parameterized simulations replicating the narrative, sensitivity and tanker water market analyses presented in the associated publications of the JWM and JWM-T can be selected by modifying the Excel file "data/excel_data/model_setup.xlsx" *(see instructions there).* Lists of these pre-parameterized simulations can be found in the overview PDF files in the subfolders "narrative_simulation_inputs/", "sensitivity_simulation_inputs/", and "tanker_simulation_inputs/" under "data/excel_data/". Customized model parameter input files can be defined in the "user_simulation_inputs/" subfolder.

The “model_setup.xlsx” also defines the length of the simulation in years, the start month, the model network, and the engine list. We caution against modifying the engine list or network without a proper understanding of the model process and code, as inconsistencies between these options and the model code can result in an error when running a simulation.

Acknowledgments:

This work product development was supported by the National Science Foundation (NSF) under Grant GEO/OAD-1342869 and Grant ICER/EAR-1829999 (under the Belmont Forum Sustainable Urbanisation Global Initiative (SUGI) / Food-Water-Energy Nexus theme). The NERC Belmont Forum provided UK funding (NE/L009285/1). As part of the Belmont Forum, the Deutsche Forschungsgemeinschaft (DFG) provided funding to the Helmholtz Centre for Environmental Research (UFZ) (KL 2764/1-1) and Leipzig University (GA 506/4-1), as well as the German Federal Ministry of Education and Research (BMBF) funding to UFZ (033WU002). The University of Manchester’s Computational Shared Facility is acknowledged. Any opinions, findings, and conclusions or recommendations resulting from in this product are solely those of the authors and do not necessarily reflect the views of the NSF or other agencies that provided funding or data.