LA-UR-23-24479

Approved for public release; distribution is unlimited.

Title: Guide to runquic.py Python Script

Author(s): Nelson, Matthew Aaron

Intended for: Report

Issued: 2023-04-27

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Guide to runquic.py Python Script

Matthew Nelson, PhD
Los Alamos National Laboratory
(505)665-3842 (Office)
(505)412-8856 (Mobile)
nelsonm@lanl.gov

The runquic.py Python script uses Python 3.8 or newer and requires several Python packages to be installed including: NumPy, SciPy, Pandas, Geopandas, and Shapely. The script assumes that the batch QUIC of project folders with all of the input files (except for any input files interpolated from WRF output files) have already be saved to the drive and are ready to be run. If the project is set to use WRF model output, it will be interpolated at runtime. Below is a list of the required and optional key word and runtime arguments used to specify for the runquic.py script:

Required Runtime Input keywords:

`proj` – The fully qualified path for the main QUIC project directory, i.e., the directory where the *.proj file and any project subdomain directories are located.

Note: that runquic.py assumes that all the required QUIC input files (*.inp) are named and formatted correctly in the proper subdirectories within the specified project directory. QUIC project directories produced by QUICGUI should meet this requirement automatically. Care must be taken to ensure that all the dependencies are met when creating QUIC projects without the QUICGUI. Some options have some obscure dependencies, such as running QUIC-PRESSURE requires the staggered velocity field flag to be 1 in QU_fileoptions.inp.

`exe` – The fully qualified path to the “executables” directory included with your installation of QUIC where all the Fortran executables are located.

`op` – This is a flag that indicates which Fortran executables should be run (1 = only QUIC-URB, 2 = QUIC-URB and QUIC-PLUME, 3 = QUIC-URB, QUIC-PLUME, and QUIC-PRESSURE, 4 = only QUIC-PLUME, 5 = only QUIC-PRESSURE).

`th` – The number of threads that should be used when setting the OMP_NUM_THREADS environment variable.

Optional Runtime Inputs:

`mv` – Moving Inner Grid Flag indicating whether to remove redundant wind speed, wind direction, ship speed, and ship direction combinations from the input files. This reduces unnecessary computations and therefore speeds up runtime. Use a value of 1 to turn this option on.

wrf – The fully qualified path to the directory where the WRF model output NetCDF data files are located so they can be interpolated for use in the QUIC simulation

sp – The fully qualified path to the directory where the SciPuff executables are located so they can be used to run the dispersion calculations on the outer grid.

pf – Defines the significant particle fraction for subsequent iterations, i.e., inner grids with less than this fraction of total particles will not be run in the next iteration.

contam – This flag will cause the Python script to export CONTAM's WPC (Wind, Pressure, Concentration) file using observation points. Note that this flag should be used in conjunction with run option 3 in order to produce all of the data files required by the WPC file.

Below are some examples of calls to the runquic.py python script depending on which options (if any) are desired. Put quotes around any paths in the input arguments in order to avoid issues with spaces in directory names. In these examples, runquicDir is the path to the directory where the runquic.py and supporting Python scripts are located.

For a simple QUIC project that does not include any WRF inputs or calls to SciPuff for the outer grid, the script is called as follows to run both QUIC-URB and QUIC-PLUME using 4 threads for the simulations:

```
python ‘runquicDir/runquic.py’ proj ‘projectDirectory’ exe
‘executableDirectory’ op 2 th 4
```

If one is running a moving ship simulation and wishes to reduce the inner grid wind calculations as much as possible (redundant cases are not run) and export results into a CONTAM WPC file; the call to the Python code would be as follows:

```
python ‘runquicDir/runquic.py’ proj ‘projectDirectory’ exe
‘executableDirectory’ op 2 th 4 mv 1 contam
```

If WRF NetCDF files are to be used to define the meteorology of the QUIC simulations the call above is modified as follows:

```
python ‘runquicDir/runquic.py’ proj ‘projectDirectory’ exe
‘executableDirectory’ op 2 th 4 wrf ‘wrfDir’
```

If one wishes to use SciPuff on the outer grid of a nested QUIC simulation runquic.py would be called as follows:

```
python ‘runquicDir/runquic.py’ proj ‘projectDirectory’ exe
‘executableDirectory’ op 2 th 4 wrf ‘wrfDir’ sp ‘scipuffDir’
```