User manual: MATLAB implementation of the QSQH synthetic model of near-wall turbulence

Yunjiu Yang

December 21, 2023

Abstract

This manual describes a resource allowing to easily use the Quasi-Steady-Quasi-Homogeneous (QSQH) theory, which describes the dominant mechanism (or one of the dominant mechanisms) by which large-scale turbulent motions affect the turbulent flow in the immediate vicinity of the wall. The QSQH synthetic model is a synthetic turbulence model that for a given large-scale component of the wall friction generates a near-wall turbulent velocity field as described by the QSQH theory. Its MATLAB implementation described in this manual is a combination of a MATLAB code and an internal database. It accepts a user-provided input of the wall friction and produces an output database consisting of samples of a velocity field, the statistical properties of which are that of the field predicted by the QSQH theory on the basis of the statistics of the large-scale component of the wall friction represented by the input. The code corresponds to the up-to-date version of the QSQH theory (Chernyshenko, S. 2021, J. Fluid Mech., **916**, A52, doi.org/10.1017/jfm.2021.180). Two currently provided internal databases allow the maximum output box of streamwise $\times$ spanwise length $\approx 200 \times 120$ wall units, with the upper side of the box at about 100 wall units from the wall. The flow statistics as predicted by the QSQH theory can be easily calculated by standard tools without the need of any QSQH derivations per se.

1 Introduction

1.1 System Requirements

- Environment: MATLAB 2023b or later.
- RAM: at least 16 GB.
- Disk Space: at least 25 GB

1.2 Installation

1. Download the "`QSQH_tool.rar`" from Github, and the "`test_dataset.rar`" from Google drive.
2. Extract the files and folders in these two rar files to the same folder. This folder will be the main folder of the QSQH tool.
3. Check that the files

- `Interface.m`
- `Excute_code.m`
- `InputParameters.txt`

are in the main folder, and

- `QSQH_model_1D.m`
- `QSQH_model_3D.m`
- `y_ref.mat`.
- `RNGseed.mat`.

are in the subfolder "`QSQH_model`". These files are the main body of the QSQH tool.

The "`progressbar.m`" script, which will also be in the folder, has been downloaded from MATLAB FileExchange. (Steve Hoelzer (2023). progressbar (`https://www.mathworks.com/matlabcentral/fileexchange/6922-progressbar`), Retrieved December 19, 2023.)

4. The main folder should also contain the subfolders:

   - `input_friction_field_test`
   - `output_1D_test`
   - `output_3D_test`
   - `universal_velocity_field_1D_test`
   - `universal_velocity_field_3D_test`

   These are the test version of the input friction field and the universal velocity field, as well as the reference output, which is used to validate the test output.

5. There are two functions of the tool in the main folder:

   - `Stats_calculator.m`
   - `loop_calculator.m`

   Their features will be described later.

2 Test of the installation

A test is designed to check the main body of the QSQH tool. The current version of QSQH tool allows to generate 1D and 3D output by running two individual codes, "`QSQH_model_1D.m`" and "`QSQH_model_3D.m`" respectively. This test will check these two codes separately.

1. Launch MATLAB. Open the file "`InputParameters`". Check that the parameters are exactly as below:

   - `Lx,2.513274122871834e+04`
   - `Lz,9.424777960769379e+03`
   - `xmax,0`
   - `zmax,0`
   - `ymin,0`
   - `ymax,50`
   - `test,true`

- `starting_loop,8`
- `ending_loop,8`
- `file_separator,\`

These are the input parameters for the 1D code.

2. Make sure the MATLAB current folder is the main folder. Open and run "`Interface.m`". Wait until the end of the calculation. On an average laptop it should take about 2 minutes. The step will let "`QSQH_model_1D.m`" generate the 1D output.

3. Open the file "`InputParameters`" again. Only change the the following parameters:

   - `xmax,204`
   - `zmax,120`
   - `ymax,5`
   - `starting_loop,4`
   - `ending_loop,4`

   The input parameters are now set for the 3D code.

4. Run "`Interface.m`" again. Wait until the end of the calculation. On an average laptop it should take about 15 minutes. The step will let "`QSQH_model_3D.m`" generate the 3D output.

5. Open "`Stats_calculator.m`" and press Run.

6. Check the resulting plots of the mean velocity and the root mean square of the fluctuation velocity obtained by the calculation together with the reference data, and a plot of two point correlations with the reference data. If they coincide, the installation was successful.

The reason of implementing the 1D and 3D QSQH model separately by two codes, is due to the different complexity of coding of implementing the 1D and 3D model. In general, the code of 1D model is much faster than the 3D model, provided that they generate the same number of instances on the same computing device.

3 Using the tool

Both the input data and the universal velocity fields used in Section 2 were for testing the installation. They should be replaced with the working versions.

A universal velocity field file is a representation of the hypothetical 'universal' near-wall velocity field that is independent of the Reynolds number, large-scale geometry of the flow and the large-scale motions, as described by the QSQH theory. Two such files of 1D and 3D universal velocity field are currently available as a part of the implementation of the QSQH synthetic model. The data in both files are obtained from the same base flow of $\mathrm{Re}_\tau = 5200$ produced by Lee and Moser [2015], accessible at John Hopkins Turbulence Database. The 1D file is explicitly for the generation of 1D output, while the 3D file is explicitly for the generation of 3D output. The files can be simply downloaded and will not require any further effort from the user:

- 1D universal velocity field.
- 3D universal velocity field.

Extract the folders in the `rar` files to the main folder of the QSQH tool. More universal field data files might be provided in the future.

The input data, however, should be prepared by the user specifically for the flow of interest. The velocity field of this flow must be statistically homogeneous in x and z directions. The input data must be the wall friction field of the flow of interest. The user needs to obtain the snapshots of the wall shear stresses $\tau_x^+(x^+, z^+)$ and $\tau_z^+(x^+, z^+)$ in wall units, from the time series of this flow. The snapshots is desired to have huge time interval between each other, for example, approximately one flow-through time.

The snapshots of $\tau_x^+(x^+, z^+)$ and $\tau_z^+(x^+, z^+)$ must be stored in the form of MATLAB matrix files of the size of $N_x \times N_z$, where N_x and N_z are the number of grid points in stream-wise and span-wise directions of the wall friction field. The user needs to take a note of the size of the stream-wise and span-wise domain in wall unit, L_x^+ and L_z^+ for later use. The snapshots must not be cropped in (x, z) directions. The matrix files should be named as: "`tau_xxxxx.mat`" where "`xxxxx`" stands for the time step or the index of the snapshot. The snapshots should be sufficient so that the friction field is statistically representative. The number of snapshots denoted N.

Create a new subfolder "`input_friction_field`" in the main folder. Store the files of the snapshots of the friction field in this subfolder. Also create two new subfolders "`output_1D`" and "`output_3D`" in the main folder for storing the output.

3.1 Input parameters

Apart from the input data of the friction field, the input parameters of the tool must be changed accordingly to the input data and the universal velocity field used. The input parameters of the tool are stored in "`InputParameters.txt`". Open this `txt` files in MATLAB. On the left of the comma, there are the names of the input parameters, and on the right, showing the values of these input parameters.

`InputParameters.txt`:

- `Lx` stands for the size of the stream-wise domain L_x^+ of the input friction field in wall unit.
- `Lz` stands for the size of the span-wise domain L_z^+ of the input friction field in wall unit.
- `xmax` stands for the user-defined size of the stream-wise domain x^+_{max} of the output synthetic velocity field in wall unit. Set $x^+_{\text{max}} = 0$ for the generation of 1D output. Set it to be positive integer of the multiple of 12 for 3D output.
- `zmax` stands for the user-defined size of the span-wise domain z^+_{max} of the output synthetic velocity field in wall unit. Set $z^+_{\text{max}} = 0$ for 1D model. Set it to be positive integer of the multiple of 6 for 3D output. Read the guidance in this section for more information about x^+_{max} and z^+_{max}.
- `ymin` stands for the user-defined closest wall-normal position to the wall y^+_{min} of the output synthetic velocity field in wall unit. You are free to choose within the wall-normal range of $y^+ \in \{0, 100\}$.
- `ymax` stands for the user-defined outermost wall-normal position y^+_{max} of the output synthetic velocity field in wall unit. You are free to choose within the wall-normal range of $y^+ \in \{0, 100\}$. The user must follow $y^+_{\text{max}} > y^+_{\text{min}}$.
- `test` tells the model whether this upcoming run is for the test of the installation, the file names of the output for the test will be marked with the ending "`test`". Set it to true for the test, or false for other runs.
- `starting_loop` stands for the starting loop of the calculation, denoted b_{start}.

```
There are up to 11 loops, the maximum number of output instances is 3145728.
Loop 1 generates 4 output instances.
Loop 2 generates 16 output instances.
Loop 3 generates 64 output instances.
Loop 4 generates 256 output instances.
Loop 5 generates 1024 output instances.
Loop 6 generates 4096 output instances.
Loop 7 generates 16384 output instances.
Loop 8 generates 65536 output instances.
Loop 9 generates 262144 output instances.
Loop 10 generates 1048576 output instances.
Loop 11 generates 3145728 output instances.
```

Figure 1: Information of loops displayed by the function.

- `ending_loop` stands for the ending loop of the calculation, denoted b_{end}. The user must follow $y^+_{\text{end}} > y^+_{\text{start}}$. Read the guidance for more information about b_{start} and b_{end}.
- `file_separator` stands for the file separator of the system. For example, it is "\" for Windows system, and "/" for the Linux system.

Guidance: the values of $x^+_{\max}$ and $z^+_{\max}$ are limited by the universal velocity field implemented in the tool. For the current version of the QSQH tool, the user is provided with the choices within the range $x^+_{\max} \in \{0, 204\}$ and $z^+_{\max} \in \{0, 120\}$. More choices would be available if new universal velocity field is provided in the future.

The current version of the QSQH tool does not allow the user to define the number of output instances. The tool generates incremental output instances in iterative loops, where more output instances are generated for higher loop number. Thereby, the number of output instances can be controlled by setting the loop number. The tool provides the user with a function to determine the relationship between the loop number and the number of output instances, before running the QSQH model. Follow the steps to use this function:

1. Open file "`InputParameters.txt`". Set the parameters `test` and `file_separator`.
2. Make sure the MATLAB current folder is the main folder. Open and run "`loop_-calculator.m`" in MATLAB. Wait until the calculation finishes. Check the command window in MATLAB.

The total number of loops and the maximum possible number of output instances are displayed. The explicit number of output instances generated for each loop is also displayed. Figure 1 is an example of the information displayed by this function. There are two ways of using this information. First, set b_{start} and b_{end} to be the same loop number. In this case, only one loop will be calculated, and a single file containing certain number of output instances will be generated. Second, set b_{start} and b_{end} differently. The QSQH model will be run for $(b_{\text{end}} - b_{\text{start}})$ loops, and generate $(b_{\text{end}} - b_{\text{start}})$ files containing different number of output instances.

3.2 Output of the model and post-process

The output files are in the form of "`.mat`" files. The output of the 1D QSQH synthetic model will be the instances of the synthetic velocity profiles in the wall-normal direction, where the outermost position is at $y^+_{\max}$ as defined in "`InputParameters_1D.txt`". The output matrix files are of the size: $N^* \times N^*_\tau \times N^*_y$, where N^*_t is the number of snapshots of the input friction field, and N^*_τ is the number of points sampled from each snapshot of the friction field, therefore, $N^*_t \times N^*_\tau$ yields the number of instances generated by the model. Each instance has the wall-normal grid that

$$y^+_{\min} < y^+ < y^+_{\max}$$

with N_y^* grid points. The wall-normal grid is obtained from the grid of the base flow of the universal velocity field, which is implemented in the current tool. The base flow is available at Johns Hopkins Turbulence Databases.

The output of the 3D QSQH synthetic model will be the instances of the synthetic velocity field which has the grid of the size:

$$-x^+_{\text{max}}/2 < x^+ < x^+_{\text{max}}/2$$
$$y^+_{\text{min}} < y^+ < y^+_{\text{max}}$$
$$-z^+_{\text{max}}/2 < z^+ < z^+_{\text{max}}/2$$

with the grid spacing $\Delta x^+ = 12$ and $\Delta z^+ = 6$. The wall-normal grids of the 1D and 3D output are the same. The output matrix files are of the size: $N^* \times N_\tau^* \times N_x^* \times N_z^* \times N_y^*$, where $N_x^* \times N_z^* \times N_y^*$ are the grid points of the synthetic velocity field in x, z and y directions.

The file "`Stats_calculator.m`" allows to calculate some basic statistics such as mean velocity profile, squared root mean square of the velocity fluctuation and two-point correlation coefficients in (x, z) directions, of the output synthetic velocity field. The functions and features of this file can be further derived to produce more flow statistics.

Due to the large scales in the input friction field, the 3D QSQH model has to implement the expand/compress of the grid of universal velocity field in (x, y, z) directions, and rotation of the grid, as described by the QSQH theory. The synthetic velocity field is obtained by interpolating the expanded/compressed and rotated universal velocity field onto a user-defined grid $x^+_{\text{max}} \times z^+_{\text{max}} \times (y^+_{\text{max}} - y^+_{\text{min}})$. No extrapolation will be done if the points of the user-defined grid fall out of the distorted grid, hence there are potentially "`NaN`" at the edge of the user-defined grid for some output instances, when the distorted grid suffers from extreme expansion/compression and rotation. The tool provides the user with the function of calculating and plotting the probability of the appearance of none-`NaN` on the (x, z) positions of the output field. Open file "`Stats_calculator.m`", go to the bottom of the code, and uncomment the last section which is responsible for this function. Figure 2 is an example of using this function on the reference output that was introduced for the test of installation.

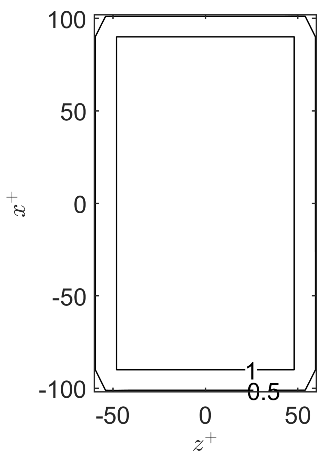

Figure 2: Probability of meaningful data point among all the (x, z) grid points of the reference output.

The area covered by the level of 1 is the area containing no `NaN`, which is:

$$-90 < x^+ < 90$$
$$-48 < z^+ < 48$$

On the edge of the grid, there is a 50% chance for "`NaN`" to appear.

References

M. Lee and R. D. Moser. Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$. *Journal of Fluid Mechanics*, 774:395–415, 2015. doi: 10.1017/jfm.2015.268.