Instruction for the LCS Matlab Kit V2

About LCS Matlab Kit

This MATLAB software package was developed in the Biological Propulsion Laboratory at California Institute of Technology. It enables users to input a time-series of 2-D velocity field data (e.g., DPIV measurements or CFD calculations) and compute the corresponding finite-time Lyapunov exponent (FTLE) fields, from which Lagrangian Coherent Structures (LCS) such as vortices and fluid transport barriers can be identified. Before reading the following instruction, please see Shadden et al. (2006) or the LCS Online Tutorial for background and technical details.

Requirement for operation system and Matlab

This software kit requires Matlab 7.0 or higher version to run. No special toolbox is required. It has been tested in Windows XP and Mac OS X. Since all the codes are written in Matlab language, it is assumed that it is able to run in other systems as long as Matlab is installed.

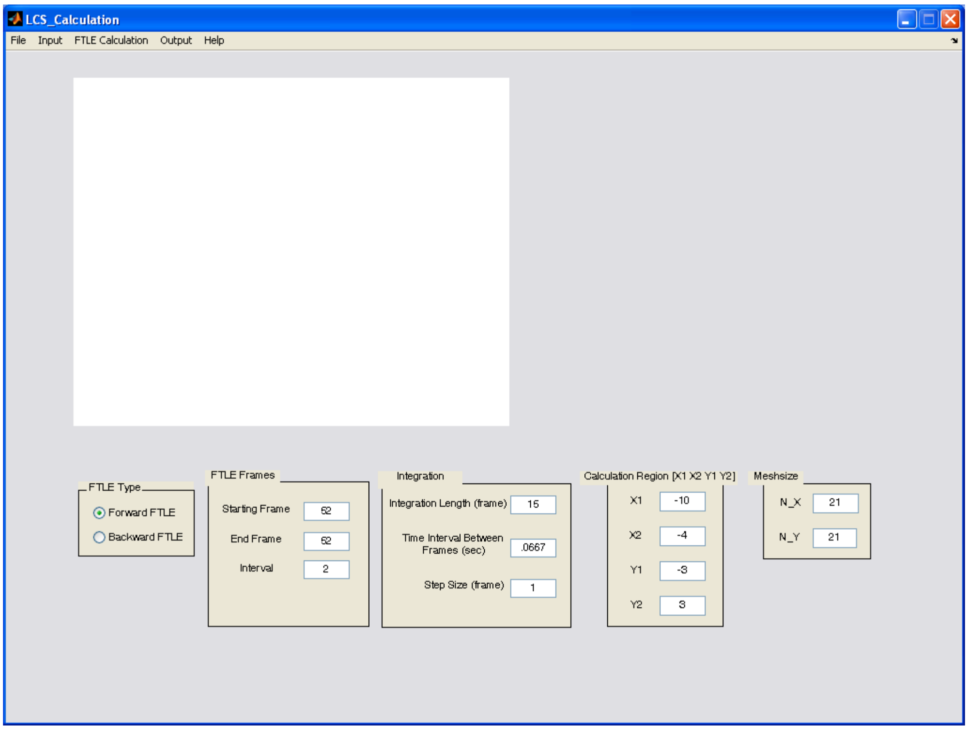
How to Use

Step 1. Start the software

1.1 Run Matlab;

1.2 Open LCS_Calculation.m and run it;

1.3 The Graphic User Interface (GUI) shows like Fig. 1.1.



FTLE Type
● Forward FTLE
○ Backward FTLE

FTLE Frames
Starting Frame: 52
End Frame: 52
Interval: 2

Integration
Integration Length (frame): 15
Time Interval Between Frames (sec): .0667
Step Size (frame): 1

Calculation Region [X1 X2 Y1 Y2]
X1: -10
X2: -4
Y1: -3
Y2: 3

Meshsize
N_X: 21
N_Y: 21

Figure 1.1

Step 2. Convert original data files

Users need to load a time-series of discrete 2-D velocity data files, either from DPIV measurements or from CFD calculations. As in usual, these files are named in such a way that their names have the same prefix and an integer from an ascending number series. For example, ‘velocity00.txt’, ‘velocity01.txt’, ‘velocity02.txt’……

The current version supports velocity data files in text (.txt) and worksheet (.wk1) format. However, the data layouts in these files have to be of the following format:

a.) Each file contains velocity data on the same M×N grid. It has four columns:

Column 1: x coordinates (X);
Column 2: y coordinates (Y);
Column 3: x velocity (U);
Column 4: y velocity (V);

b.) X and Y coordinates are sorted in ascending order;

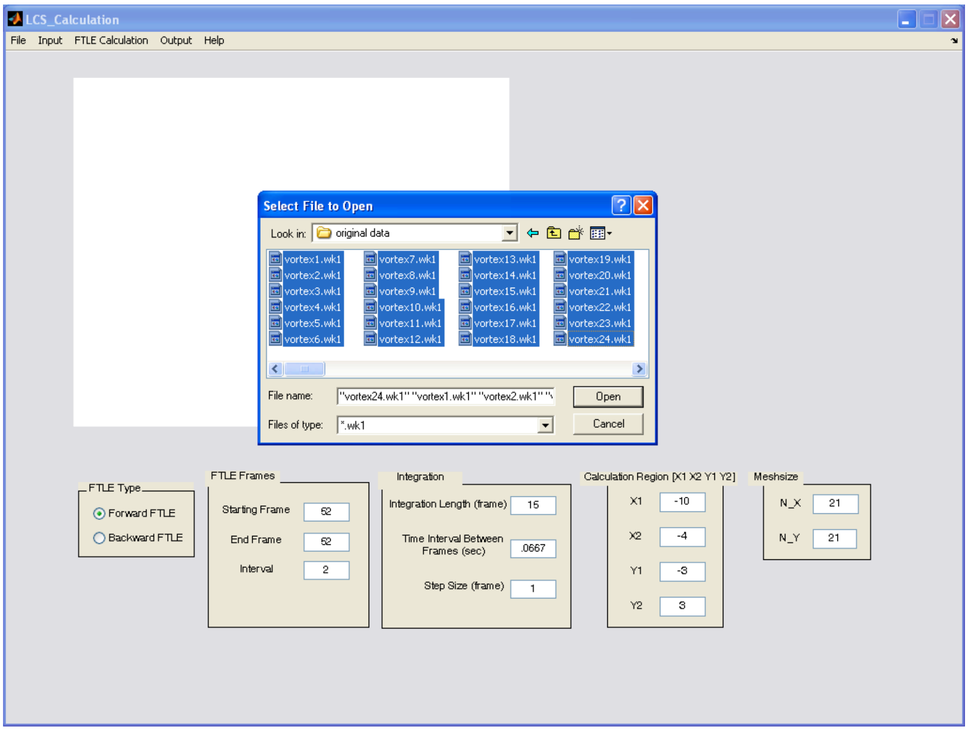
c.) Each row indicates the velocity data (U, V) at that location (X, Y).

The following is an example of the velocity file:

X_1	Y_1	U_{11}	V_{11}
X_1	Y_2	U_{12}	V_{12}
X_1	Y_3	U_{13}	V_{13}
……….			
X_1	Y_N	U_{1N}	V_{1N}
X_2	Y_1	U_{21}	V_{21}
……….			
X_2	Y_N	U_{2N}	V_{2N}
……………			
X_M	Y_1	U_{M1}	V_{M1}
……….			
X_M	Y_N	U_{MN}	V_{MN}

The followings are steps to load the velocity files:

2.1 From the GUI, select from menu ‘Input -> Load Original Data’. A ‘Select File to Open’ dialogue box appears. Select the velocity files and click ‘Open’ (Fig. 2.1).



Look in: original data

vortex1.wk1, vortex2.wk1, vortex3.wk1, vortex4.wk1, vortex5.wk1, vortex6.wk1, vortex7.wk1, vortex8.wk1, vortex9.wk1, vortex10.wk1, vortex11.wk1, vortex12.wk1, vortex13.wk1, vortex14.wk1, vortex15.wk1, vortex16.wk1, vortex17.wk1, vortex18.wk1, vortex19.wk1, vortex20.wk1, vortex21.wk1, vortex22.wk1, vortex23.wk1, vortex24.wk1

File name: "vortex24.wk1" "vortex1.wk1" "vortex2.wk1" "\

Files of type: *.wk1

FTLE Type
● Forward FTLE
○ Backward FTLE

FTLE Frames
Starting Frame: 52
End Frame: 52
Interval: 2

Integration
Integration Length (frame): 15
Time Interval Between Frames (sec): .0667
Step Size (frame): 1

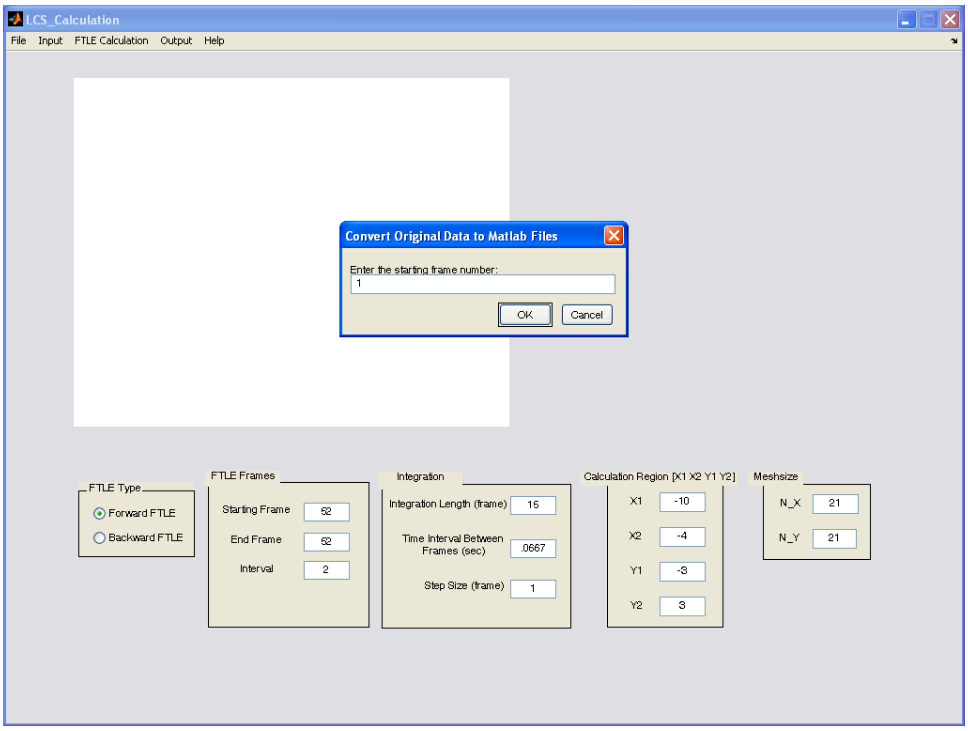
Calculation Region [X1 X2 Y1 Y2]
X1: -10
X2: -4
Y1: -3
Y2: 3

Meshsize
N_X: 21
N_Y: 21

Figure 2.1

2.2 Since the software runs in Matlab, all velocity files need to be converted into Matlab .mat files to reduce computational time (velocity files are repeatedly read in the computation). Select from menu ‘Input -> Convert to MAT files’. A ‘Browse for Folder’ dialogue box appears. Select the folder in which you want these .mat files to be saved. After clicking ‘OK’, a new dialogue box appears in which users need to enter the starting frame number. It is suggested to use the default value ‘1’. Then click ‘OK’ (Fig. 2.2).

This step would generate a series of files named ‘U_T1, U_T2….’ and a series of files named ‘V_T1, V_T2….’. All of the files are of an M×N matrix, with U_T and V_T represent X velocity and Y velocity, respectively. It is noted that no matter what numeration is used in naming the original velocity file, the converted files use the numeration ‘1, 2, 3, ….’. In addition, two files named ‘xgrid.mat’ and ‘ygrid.mat’ are generated which contains coordinates of the grid. In summary, for an M×N grid, ‘xgrid’ is an M×1 vector while ‘ygrid’ is an N×1 vector. ‘U_T’ and ‘V_T’ are M×N matrix. U_T(i,j) and V_T(i,j) indicate x and y velocity at location [xgrid(i), ygrid(j)].



Convert Original Data to Matlab Files

Enter the starting frame number: 1

FTLE Type

● Forward FTLE

○ Backward FTLE

FTLE Frames	
Starting Frame	62
End Frame	62
Interval	2

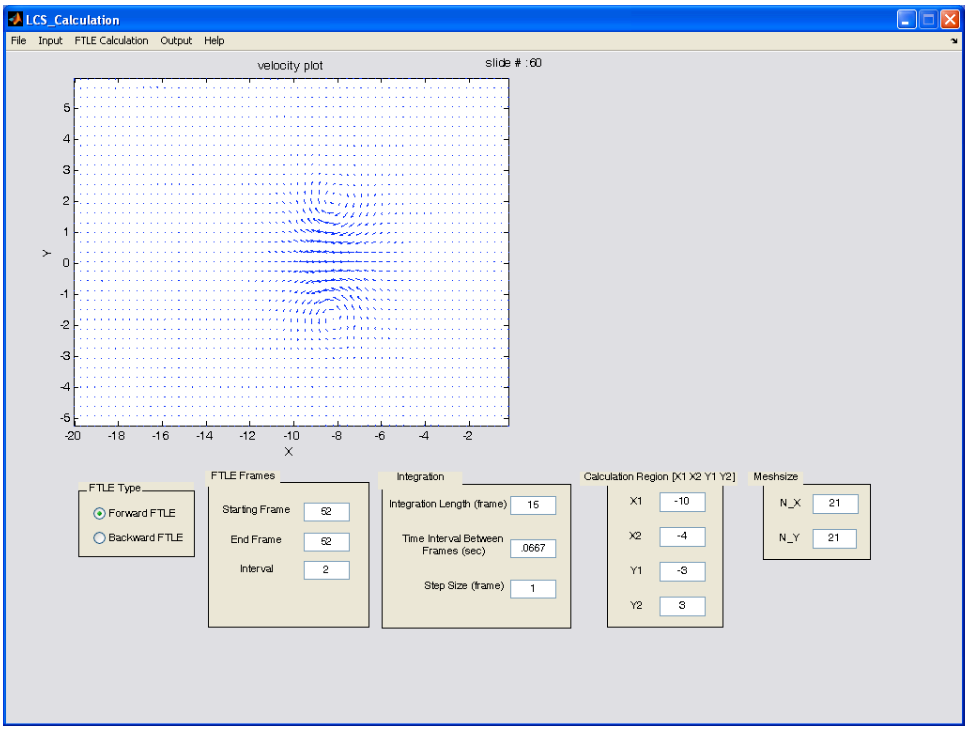
Integration	
Integration Length (frame)	15
Time Interval Between Frames (sec)	.0667
Step Size (frame)	1

Calculation Region [X1 X2 Y1 Y2]	
X1	-10
X2	-4
Y1	-3
Y2	3

Meshsize	
N_X	21
N_Y	21

Figure 2.2

2.3 The velocity can be plotted by selecting ‘Input -> Plot Velocity Field’. Enter the starting and ending frame number in the dialogue box. Then click ‘OK’ and the velocity field is plotted on the interface (Fig. 2.3). Make sure the starting and ending frames number do not exceed the number of frames.



FTLE Type

● Forward FTLE

○ Backward FTLE

FTLE Frames	
Starting Frame	62
End Frame	62
Interval	2

Integration	
Integration Length (frame)	15
Time Interval Between Frames (sec)	.0667
Step Size (frame)	1

Calculation Region [X1 X2 Y1 Y2]	
X1	-10
X2	-4
Y1	-3
Y2	3

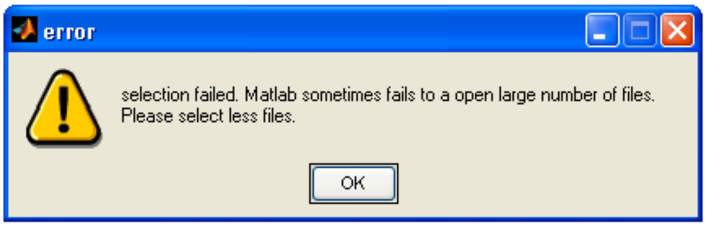
Meshsize	
N_X	21
N_Y	21

Figure 2.3

Notes:

a. For every data set, the file conversion in this step only needs to be processed once. After that every computation starts from step 3.

b. Due to an internal bug in Matlab command ‘uigetfile’, it sometimes fails to open large number of files. An error dialog (Fig. 2.4) will show in this case. So don’t try to load all the original velocity files and convert them at the same time. Just convert a portion of them and enter the correct starting frame number for every conversion. Make sure the frame numbers are continuous integers. To load and convert more files at one time, make the name for the original files as short as possible, e.g., V01.txt, V02.txt…..



selection failed. Matlab sometimes fails to a open large number of files.
Please select less files.

Figure 2.4

Step 3. Set up the parameters and calculate FTLE

All the parameters users need to assign values are shown on the GUI panel.

3.1 Specify the folder where velocity.mat files (from section 2.3) are located using menu ‘FTLE Calculation -> Load Velocity Files’. Notice that the folder, not the files in it is selected.

3.2 In the Calculation Type group box, choose Forward or Backward FTLE calculation;

3.3 In the FTLE Frame edit box, enter frame numbers where FTLE is calculated. A series of FTLE are calculated from ‘Starting Frame’ to ‘End Frame’ with ‘Interval’.

3.4 In the Integration edit boxes, enter values for integration parameters.

a.) Integration Length:

This parameter tells the time duration (in frames) over which the trajectories are integrated.

b.) Time interval between consecutive frames:

This is the actual time interval (in seconds) between two consecutive frames.

c.) Step size:

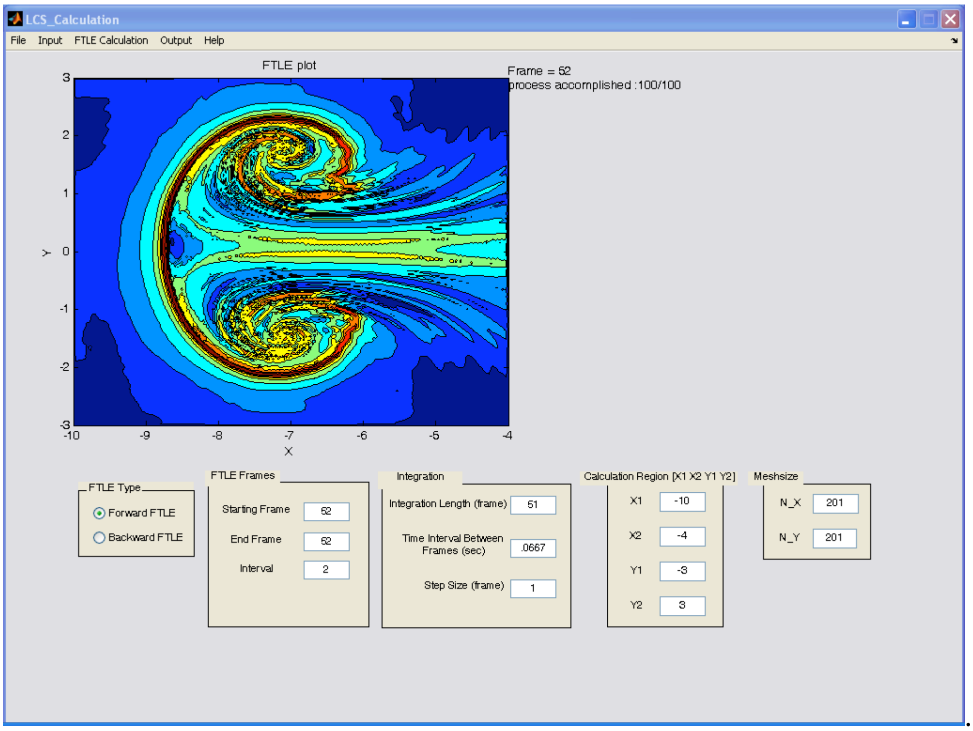
Number of frames per integration step. Default to 1.

3.5 Calculation region and mesh size

[X1 X2 Y1 Y2] is the computational domain over which the FTLE is calculated. It must be a sub-domain of the velocity field. In this domain, the FTLE is calculated on nodes of a N_X×N_Y grid. Larger N_X and N_Y give result with high resolution, but the computation cost is higher. Generally, use a larger computational domain with a course grid to get some rough understanding of LCS in initial trials, then limit the domain to where the LCS is located and adopt a denser grid.

3.6 Calculate FTLE

Click menu ‘FTLE Calculation -> Compute FTLE’ to start the calculation. FTLE is calculated and plotted on the GUI panel (Fig. 3.1)



Frame = 52
process accomplished :100/100

FTLE Type
● Forward FTLE
○ Backward FTLE

FTLE Frames
Starting Frame: 52
End Frame: 52
Interval: 2

Integration
Integration Length (frame): 51
Time Interval Between Frames (sec): .0667
Step Size (frame): 1

Calculation Region [X1 X2 Y1 Y2]
X1: -10
X2: -4
Y1: -3
Y2: 3

Meshsize
N_X: 201
N_Y: 201

Figure 3.1

Step 4. Save the result.

Click menu ‘Output -> Save FTLE Calculation’ to save the result. The saved result is a text (.txt) file with Tecplot contour plot heading. FTLE at different frames are outputted into separate zones. It can be directly loaded into Tecplot.

Acknowledgement

Great thanks to Shawn Shadden, whose work provides theoretical background and numerical algorithm of this software kit.

References

1. Shadden, S. C., Dabiri, J. O. and Marsden, J. E. (2006) Lagrangian analysis of entrained and detrained fluid in vortex rings. *Phys. Fluids* **18**, 047105.
2. Lagrangian Coherent Structures: Analysis of time-dependent dynamics system using finite-element Lyapunov exponents. http://www.cds.caltech.edu/~shawn/LCS-tutorial/