

# Appendix A

## Newman - Software for Computing FTLE in Geophysical Applications

**Newman** is software to compute the FTLE for time-dependent velocity fields. The velocity field is provided either as an analytical expression, or as a numerical data set. The computations can be performed on parallel processors or on a single processor.

### A.1 Installation

**Newman** source code can be compiled using the make command. To install **Newman** you will need to complete the following three steps:

1. Install the MPI library,
2. Install the GSL library,
3. Make **Newman**,

The MPI and GSL libraries must be installed before **Newman** can be compiled.

#### A.1.1 MPI

**Newman** is designed for industrial strength computations in parallel. Computations can be run on a single processor; however, MPI must still be installed. Visit [http:](http://)

`//www.open-mpi.org` to find instructions for downloading and installing the MPI library.

### A.1.2 GSL

`Newman` uses the integration and eigensystem routines in the GNU Scientific Library. Visit <http://www.gnu.org/software/gsl/> to find instructions for downloading and installing the GSL library. `Newman` requires version GSL 1.10 or greater. To determine which version of GSL you currently have and where it is installed, simply execute the `gsl-config` command.

### A.1.3 Make Newman

The `Newman` source code resides in the `./src` directory. A shell script is provided that will attempt to compile `Newman` in two, three, and four dimensions. Make sure that the permissions on the script `./makenewman.sh` are set to executable, and then run the script:

```
>./makenewman.sh
```

This will attempt to generate three executable files: `newman2D`, `newman3D`, and `newman4D` in the `./src` directory.

If there are errors in compilation, or if you would like to customize the installation, you may need to edit the makefiles. The makefiles can be found in the `./src` directory. The most common error encountered when compiling is ensuring that the linker can find the needed libraries.

## A.2 Running Newman

To run `Newman` from the commandline, type

```
>newman2D saturn.in
```

where saturn.in is a text input file containing parameter values for your computation. Be sure that the executable is in a directory in your path.

See the Examples directory for an example of an input file. A description of each of the parameters and their default values is provided in section A.4.

## A.3 Viewing the Results

Newman outputs all results in a lightweight binary format. Separate code modules are then used to convert the raw binary format into the final desired format. This allows for:

- fast retrieval of results from a cluster over the internet,
- easier writing of modules for data conversion to the final desired format required by the user.

### A.3.1 Tecplot Format

The `raw2tec` modules are provided to convert the raw binary output from Newman to Tecplot binary format. Tecplot is commercial plotting software. To compile the `raw2tec` modules, run the `makeraw2tec.sh` shell script.

```
>./makeraw2tec.sh
```

This script will attempt to compile `raw2tec2D`, `raw2tec3D`, and `raw2tec4D` in the `./raw2tec` directory. The first step executed by the script will be to compile the `tecio.a` library in the `./lib/tecio/` directory. The `tecio.a` library is written by the Tecplot company. As the compilation proceeds, you will be requested to select your platform. I have found that selecting `linux.22` as the platform works well on Unix and Mac platforms. Next, when directed, select the option: “1. Make `tecio.a` only”. The

script should then proceed uninterrupted to the finish. The raw2tec executables will be in the ./raw2tec directory.

To learn more about these modules, type

```
>raw2tec2D -h
```

at the commandline.

The binary data can be formatted for plotting in Cartesian coordinates or on the sphere for geophysical applications. Use the `-s` flag to convert from spherical coordinates to cartesian for plotting on the sphere.

For 3D data on the sphere, use the `-a` option to provide the scale factor by which to scale the atmosphere with respect to its true size. For example, `-a 20.0` will scale the atmosphere by a factor of 20.0 relative to the sphere radius.

### A.3.2 Matlab Format

In the ./scripts directory, you will find the raw2mat.m Matlab script. This script allows for conversion of the raw binary output to Matlab form. The Matlab conversion script raw2mat is not as well supported as the Tecplot conversion module raw2tec. Many of the features available in raw2tec still need to be included in the raw2mat.m script.

## A.4 Parameters

### **Atmos\_Radius**

Values: double

Default: 6378.1

Description: Atmos\_Radius is the radius of the sphere to be used for integration on the sphere.

### **Atmos\_Set**

Values: [ 0 | 1 ]

Default: 0

Description: Atmos\_Set is a switch to turn on/off integration on the sphere using Longitude and Latitude coordinates.

#### **Boundary\_InFile**

Values: char\*

Default: boundary.dat

Description: Boundary\_InFile is the name of the file containing the list of polygon segments to be used for the boundary.

#### **Boundary\_Method**

Values: [ 0 | 1 | 2 ]

Default: 0

Description: Boundary\_Method determines the type of boundary checking that will be used according to the following table:

0 No enforcement of boundaries.

1 Regions outside the boundary are defined as those for which the velocity data cell contains a value that is NaN.

2 The boundary is explicitly provided as a list of polygon segments. 3 An analytical expression hard-wired into the code is evaluated to determine the boundary.

#### **Data\_Format**

Values: [ 0 | 1 | 2 ]

Default: 2

Description: Data\_Format determines the format of the numerical velocity data according to the following table:

0 Raw binary data

1 Tecplot ASCII data

2 ASCII lean velocity data

#### **Data\_InFile**

Values: char\*

Default: velocitydata

Description: Base name of the files containing the numerical velocity data.

#### **Data\_Max[ND]**

Values: double

Default: 1.0

Description: Data\_Max[ND] is the maximum value of grid locations in the numerical velocity data.

#### **Data\_Min[ND]**

Values: double

Default: 0.0

Description: `Data_Min[ND]` is the minimum value of grid locations in the numerical velocity data.

**`Data_NonUniformGridND`**

Values: [ 0 | 1 ]

Default: 0

Description: `Data_NonUniformGrid[ND]` is a switch to turn on/off the use of a non-uniform grid for the locations of numerical velocity data along the given dimension.

**`Data_NumInputFiles`**

Values: int

Default: 2

Description: Number of input files containing the numerical velocity data.

**`Data_Periodic[ND]`**

Values: [ 0 | 1 ]

Default: 0

Description: `Data_Periodic[ND]` is a switch to turn on/off periodicity of the data along the given spatial dimension.

**`Data_Res[ND]`**

Values: int

Default: 2

Description: `Data_Res[ND]` is the number of grid locations in the numerical velocity data.

**`Data_TDelta`**

Values: double

Default: 1.0

Description: `Data_TDelta[ND]` is the time interval between time slices in the numerical velocity data.

**`Data_TMin`**

Values: double

Default: 0.0

Description:

**`Data_TPeriodic`**

Values: [ 0 | 1 ]

Default: 0

Description: `Data_TPeriodic` is a switch to turn on/off time periodicity in the numerical velocity data.

**`Data_TRes`**

Values: int

Default: 2

Description: Data\_TRes is the number of time slices in the numerical velocity data.

**Filter\_Width**

Values: double

Default: 0.0

Description: FTLE\_Width is the width of the gaussian filter to be used in post process smoothing of the FTLE.

**FTLE\_Compute**

Values: [ 0 | 1 ]

Default: 1

Description: A switch to turn on/off computation of the Finite Time Liapunov Exponent field.

**FTLE\_IntTLength**

Values: double

Default: 1.0

Description: FTLE\_IntTLength is the integration time to be used in the definition of the FTLE.

**FTLE\_Max[ND]**

Values: double

Default: 1.0

Description: FTLE\_Max[ND] is the maximum grid location for computation of the FTLE.

**FTLE\_Min[ND]**

Values: double

Default: 0.0

Description: FTLE\_Min[ND] is the minimum grid location for computation of the FTLE.

**FTLE\_OutFile**

Values: char\*

Default: repatt

Description: FTLE\_Outfile is the base name of the output file for FTLE data.

**FTLE\_Res[ND]**

Values: int

Default: 3

Description: FTLE\_Res[ND] is the number of grid locations for computation of the FTLE.

**FTLE\_TrackWidth[ND]**

Values: double

Default: 1.0

Description: FTLE\_TrackWidth[ND] is the width of the grid for computation of the FTLE centered about the storm center when the Track\_Storm option is used.

**Int\_AbsTol**

Values: double

Default: 1e-5

Description: `Int_AbsTol` is the tolerance threshold for the absolute error for an adaptive step size integrator.

#### **Int\_MaxTimeStep**

Values: double

Default: 1.0

Description: `Int_MaxTimeStep` is the maximum allowed timestep for the RK45 integrator with adaptive step-sizing and step-size thresholds. The algorithm attempts to choose the largest step size between `Int_MinTimeStep` and `Int_MaxTimeStep` such that the error tolerances are satisfied.

#### **Int\_Method**

Values: [ 0 | 1 | 2 | 3 ]

Default: 0

Description: `Int_Method` determines the type of integrator used to integrate trajectories according to the following table:

0 Adaptive step size integrator with strict error tolerances

1 Fixed step size integrator

2 Euler integrator

3 RK45 integrator with adaptive step-sizing and step-size thresholds.

#### **Int\_MinTimeStep**

Values: double

Default: 0.1

Description: `Int_MinTimeStep` is the minimum allowed timestep for the RK45 integrator with adaptive step-sizing and step-size thresholds. The algorithm attempts to choose the largest step size between `Int_MinTimeStep` and `Int_MaxTimeStep` such that the error tolerances are satisfied.

#### **Int\_RelTol**

Values: double

Default: 1e-6

Description: `Int_RelTol` is the tolerance threshold for the relative error for an adaptive step size integrator.

#### **Int\_TimeStep**

Values: double

Default: 1.0

Description: `Int_TimeStep` is the size of the timestep for a fixed step size integrator.

#### **MapCoord**



Values: [ 0 | 1 ]

Default: 0

Description: A switch to turn on/off mapping of points to an integration space before integration of trajectories. This is convenient if the FTLE is to be plotted in a space that is different from the space in which the trajectories are to be integrated.

#### **Nest\_NumNests**

Values: int

Default: 0

Description: Nest\_NumNests is the number of total nests available for nested velocity grids.

#### **Nest\_List**

Values: { int }

Default: 0

Description: Nest\_List is an ordered list specifying which of the nested velocity sets should be used for the computation. Elements in the list can be any integer from 0 to Nest\_NumNests-1 inclusive.

#### **Output\_TDelta**

Values: double

Default: 1.0

Description: Output\_TDelta is the desired time interval between frames of output data.

#### **Output\_TRes**

Values: int

Default: 2

Description: Output\_TRes is the desired number of frames of output data.

#### **Output\_T1**

Values double

Default: 0.0

Description: Output\_T1 is the desired time of the first frame of output data.

#### **Parallel\_LoadRatio**

Values double

Default: 1.0

Description: Parallel\_LoadRatio is the factor by which the number of jobs to be assigned is larger than the number of processors available. Choosing a larger ratio can help to improve load balancing among the nodes at the expense of a nominal overhead.

#### **Parameters\_Print**

Values: [ 0 | 1 ]

Default: 1

Description: A switch to turn on/off printing of all parameter values to the screen.

**Path\_Input**

Values: char\*

Default: Input

Description: Path\_Input is the directory in which all input files can be found.

**Path\_Output**

Values: char\*

Default: Output

Description: Path\_Output is the directory where all output data generated will be placed.

**Path\_Work**

Values: char\*

Default: Work

Description: Path\_Work is a directory where temporary working files used by the Master node during the computation will be placed.

**Path\_Scratch**

Values: char\*

Default: /tmp

Description: Path\_Work is a directory where temporary working files used by the slave nodes during the computation will be placed. This should be a directory local to the node to avoid unnecessary transfer of files.

**Plot\_Max[ND]**

Values: double

Default: 1.0

Description: Plot\_Max[ND] is the maximum grid location for plotting of the velocity.

**Plot\_Min[ND]**

Values: double

Default: 0.0

Description: Plot\_Max[ND] is the minimum grid location for plotting of the velocity.

**Plot\_OutFile**

Values: char\*

Default: velocityplot.dat

Description: Plot\_OutFile is the name of the file where the plotted velocity output data will be placed.

**Plot\_Res[ND]**

Values: int

Default: 2

Description: Plot\_ResND] is the number of grid locations for plotting of the velocity.

**Plot\_Velocity**

Values: [ 0 | 1 ]

Default: 0

Description: A switch to turn on/off output of the velocity field for velocity data.

**Query\_Velocity**

Values: [ 0 | 1 ]

Default: 0

Description: A switch to turn on/off querying of the velocity data at a specified point.

**Query\_X[ND]**

Values: double

Default: 0.0

Description: Query\_X[ND] are the coordinates at which the velocity will be queried and printed to the screen.

**Time\_Direction**

Values: [ -1 | 1 ]

Default: 1

Description: For Plot\_Velocity and Trace computations, Time\_Direction determines the direction of time in the output data. For FTLE\_Compute, Time\_Direction determines the direction of integration of trajectories.

**Time-Origin**

Values: 6 int

Default: 0 0 0 0 0 0

Description: Time-Origin contains an array of six integers representing the time and date that correspond to time zero in the data. The integers represent: Year Month Day Hour Minute Second. Specifying the time origin allows the raw2tec utilities to time and datestamp the output for plotting purposes.

**Trace\_ColorDimension**

Values: int

Default: 0

Description: Trace\_ColorDimension is the dimension that will be used to color drifters according to their initial position.

**Trace\_Compute**

Values: [ 0 | 1 ]

Default: 0

Description: A switch to turn on/off computation of passive drifter trajectories.

**Trace\_GenerateMesh**

Values: [ 0 | 1 ]

Default: 0

Description: Trace\_GenerateMesh is a switch to turn on/off the generation of a grid of initial locations for drifter trajectories.

**Trace\_InFile**

Values: char\*

Default: drifterinput.dat

Description: Trace\_InFile is the name of the input file containing initial locations of drifters.

**Trace\_MeshColor**

Values: double

Default: 0.0

Description: Trace\_MeshColor is the color to be assigned to the grid of generated drifters.

**Trace\_MeshMax[ND]**

Values: double

Default: 1.0

Description: Trace\_MeshMax[ND] is the maximum grid location for initial positions of drifters.

**Trace\_MeshMin[ND]**

Values: double

Default: 0.0

Description: Trace\_MeshMin[ND] is the minimum grid location for initial positions of drifters.

**Trace\_MeshReleaseTime**

Values: double

Default: 0.0

Description: Trace\_MeshReleaseTime is the time at which the grid of drifters will be released in the flow.

**Trace\_MeshRes[ND]**

Values: int

Default: 2

Description: Trace\_MeshRes[ND] is the number of grid locations for initial positions of drifters.

**Trace\_OutFile**

Values: char\*

Default: drifteroutput.dat

Description: Trace\_OutFile is the name of the output file containing drifter trajectories.

**Track\_InFile**

Values: char\*

Default: track.dat

Description: Track\_File is the name of the file containing the coordinates of the track to be used for computation of FTLE for moving grids.

#### **Track\_Storm**

Values: [ 0 | 1 ]

Default: 0

Description: Track\_Storm is a switch to turn on/off computation of FTLE in storm centered coordinates.

#### **V[ND]**

Values: char\*

Default: x

Description: List of analytical equations for the velocity field.

#### **Velocity\_Format**

Values: [ 0 | 1 | 2 ]

Default: 0

Description: Velocity\_Format determines the type of velocity to be used during the computation according to the following table:

0 Equations for velocity hard wired into the code at compile time in the velocity.cpp file. 1 Equations specified by the user at run time using the equation parser. 2 Velocity is to be determined from a numerical data set.

#### **Velocity\_Null[ND]**

Values: [ 0 | 1 ]

Default: 0

Description: Velocity\_Null[ND] is a switch to turn on/off the zeroing of the velocity along the specified dimension.