

FDLyapu: Estimating the Lyapunov exponents spectrum

Sylvain Mangiarotti & Mireille Huc

2019-08-29

The **FDLyapu** package is an extension of the **GPoM** package¹.

Its aim is to assess the spectrum of the Lyapunov exponents for dynamical systems of polynomial form. The fractal dimension can also be deduced from this spectrum. The algorithms available in the package are based on the algebraic formulation of the equations. The **GPoM** package, which enables the numerical formulation of algebraic equations in a polynomial form, is thus directly required for this purpose. A user-friendly interface is also provided with **FDLyapu**, although the codes can be used in a blind mode. The connexion with **GPoM** is very natural here since this package aims to obtain Ordinary Differential Equations (ODE) from observational time series. The present package can thus be applied afterwards to characterize the global models obtained with **GPoM** but also to any dynamical system of ODEs expressed in polynomial form.

Algorithms

The Lyapunov exponents quantify the rate of separation of infinitesimally close trajectories. Depending on the initial conditions, this rate can be positive (divergence) or negative (convergence). A n -dimensional system is characterized by n characteristic Lyapunov exponents. For continuous dynamical systems, one Lyapunov exponent must correspond to the direction of the flow and should thus equal zero in average. To estimate the Lyapunov exponents spectrum, two algorithms are made available to the user, both based on the formal derivation of the Jacobian matrix (derivation is actually semi-formal only since the numeric coefficients are used in the derivation process).

⌘ Method 1 (Wolf)

The first algorithm made available in the package was introduced by *Wolf et al.* in 1985.² It is based on a Gram-Schmidt method. The algebraic formulation of the model from which the Jacobian matrix is derived is required. Considering a set of initial conditions, an ensemble of n infinitesimal perturbations is generated (one for each direction of the phase space) and the Jacobian matrix is used to estimate locally the divergence/convergence of the flow (an eigenvalue decomposition is used to distinguish the directions of the flow). Note that, in its principle, this formulation does not enable the distinction of the eigenvalue related to the direction of the flow from the other eigenvalues.

This estimation is repeated all along the flow in order to have a large ensemble of estimates. If this ensemble is large enough, the averaged values of the assessed coefficients can be considered as statistically significant for a robust estimate of the global Lyapunov exponents spectrum. Based on this algorithm, the Lyapunov exponents are organised from the largest to the smallest one, but it is not always easy to distinguish which exponent best belongs to the flow direction.

⌘ Method 2 (Grond)

¹<https://CRAN.R-project.org/package=GPoM>

²A. Wolf, J. B. Swift, H. L. Swinney, & J. A. Vastano, Determining Lyapunov exponents from a time series, *Physica D*, **16**, 285-317, 1985.

The second algorithm was introduced by *Grond et al.* in 2003.^{3 4} It is based on the same principles as method 1, except that, before applying the Gram-Schmidt method, a projection is first applied along the flow direction in order to clearly distinguish this direction from the others, enabling also a more robust estimation of all the Lyapunov exponents. One particular interest of the method is to obtain locally valid estimates of the Lyapunov exponents. Note that, due to this specific technique, the Lyapunov exponents spectrum is always organised with the zero-exponent positioned in first place. The other exponents are then organised independently from the largest to the smallest one.

⌘ The Kaplan-Yorke dimension D_{KY}

The D_{KY} dimension was introduced by *Kaplan and Yorke* in 1979.⁵ This dimension is interesting here for several reasons. First, it can be directly deduced from the Lyapunov spectrum (for this reason it is also called the Lyapunov dimension). Second, the D_{KY} dimension was proven to be robust to global modelling⁶ which is an important objective of the GPoM package to which the present package is connected. Finally, although mainly used in dimension three, its formulation is sufficiently general to be applied to higher dimensional systems. For example, it was applied to characterize the four-dimensional Ebola attractor obtained with the global modelling technique.⁷

Example

Numerical formulation of the dynamical system

To assess the Lyapunov exponents, the present approach requires the algebraic formulation of the studied dynamical system. To formulate the equations of a dynamical system, the `FDLyapu` package uses the conventions defined in the GPoM package. Details about these conventions are explained further in the vignette `GPoM : I Conventions`⁸.

The Rössler system⁹

$$\begin{aligned} dx/dt &= -y - z \\ dy/dt &= x + ay \\ dz/dt &= b + z(x - c). \end{aligned}$$

is taken here as a case study to exemplify these conventions. For $(a = 0.52, b = 2, c = 4)$, this system has a phase non-coherent chaotic behavior. Following the conventions defined in GPoM (see function `poLabs`, here with `nVar = 3` and `dMax = 2`), this system can be described by the matrix `K` as follows:

```
# parameters
a = 0.52
b = 2
c = 4
# equations
Eq1 <- c(0,-1, 0,-1, 0, 0, 0, 0, 0, 0)
```

³F. Grond, H.H. Diebner, S. Sahle, A. Mathias, S. Fischer & O.E. Rössler, A robust, locally interpretable algorithm for Lyapunov exponents. *Chaos, Solitons & Fractals*, **16**, 841-852, 2003.

⁴F. Grond & H.H. Diebner, Local Lyapunov exponents for dissipative continuous systems *Chaos Solitons & Fractals*, **23**, 1809-1817, 2005.

⁵J. L. Kaplan & J. A. Yorke, Chaotic Behavior of Multidimensional Difference Equations, in *Functional Differential Equations and Approximations of Fixed Points, Lecture Notes in Mathematics*, **730**, edited by H.-O. Peitgen & H.-O. Walter, Springer, Berlin, 1979.

⁶S. Mangiarotti, L. Drapeau & C. Letellier. Two chaotic global models for cereal crops cycles observed from satellite in Northern Morocco. *Chaos*, **24**, 023130, 2014.

⁷S. Mangiarotti, M. Peyre & M. Huc, A chaotic model for the epidemic of Ebola virus disease in West Africa (2013-2016), *Chaos*, **26**, 113112, 2016.

⁸https://cran.r-project.org/web/packages/GPoM/vignettes/b1_Conventions.pdf

⁹O. E. Rössler, An Equation for Continuous Chaos, *Physics Letters*, **57A**(5), 397-398, 1976.

```
Eq2 <- c(0, 0, 0, a, 0, 0, 1, 0, 0, 0)
Eq3 <- c(b, -c, 0, 0, 0, 0, 0, 1, 0, 0)
K = cbind(Eq1, Eq2, Eq3)
```

Indeed, using the `visuEq` function, the equations are properly formed:

```
visuEq(nVar = 3, dMax = 2, K = K, substit = c("x", "y", "z"))
## dx/dt = -z -y
##
## dy/dt = 0.52 y + x
##
## dz/dt = 2 -4 z + x z
```

(by default, the notation used in function `visuEq` is "X1", "X2" etc., the option `substit` enables here to choose an alternate notation explicitly).

Compute the Lyapunov exponent spectrum

To run the algorithm, the following elements must be provided to the algorithm:

```
# The dynamical system equations (just defined by matrix K)
#
# The number of variables (it can be deduced from matrix K)
nVar = dim(K)[2]
# The maximum polynomial degree of the formulation (also deduced from K)
pMax = dim(K)[1]
dMax = p2dMax(nVar, pMax)
# The initial conditions
inicond <- c(-0.04298734, 1.025536, 0.09057987)
# The integration time step
timeStep <- 0.01
# The initial and ending integration time
tDeb <- 0
tFin <- 10
```

The two algorithms can then be launched successively using the `lyapFDWolf` and `lyapFDGrond` functions:

```
# Prepare the output file
outLyapFD <- NULL
# Method 1
outLyapFD$Wolf <- lyapFDWolf(outLyapFD$Wolf, nVar= nVar, dMax = dMax,
                             coeffF = K,
                             tDeb = tDeb, dt = timeStep, tFin = tFin,
                             yDeb = inicond)
# Method 2
outLyapFD$Grond <- lyapFDGrond(outLyapFD$Grond, nVar= nVar, dMax = dMax,
                                coeffF = K,
                                tDeb = tDeb, dt = timeStep, tFin = tFin,
                                yDeb = inicond)
```

The output of the two algorithms will be stored here in the list `outLyapFD` which description will be given in the subsequent section.

Results

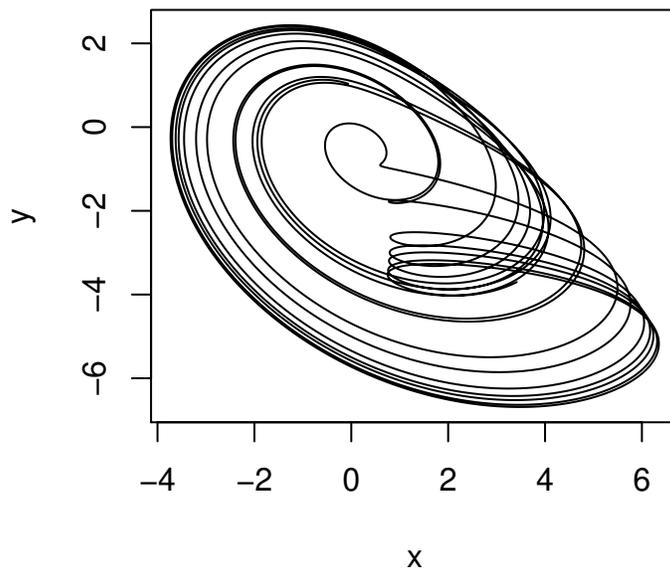
The Lyapunov exponents

The outputs of the two algorithms are organized in sublists as follows:

```
names(outLyapFD$Wolf)
## [1] "y"          "D"          "t"          "lyapExp"    "lyapExpLoc"
## [6] "SumLogNorm" "tDebInit"   "stdExp"     "meanExp"    "Dky"
## [11] "DkyLoc"     "stdDky"     "meanDky"    "methodName"
```

The trajectory in the phase space, along which the local Lyapunov exponents are computed, is stored in `$y` (the corresponding time vector is in `$t`). The corresponding phase portrait can thus be plotted as follows:

```
plot(outLyapFD$Wolf$y[,1], outLyapFD$Wolf$y[,2], type = 'l', xlab = 'x', ylab = 'y')
```



The Lyapunov exponents (locally estimated) are stored in `$lyapExpLoc` (at each iteration, the last estimate is concatenated to the previous ones). The average is estimated at each iteration and concatenated in `$lyapExp`. The last estimate of the Lyapunov exponents spectrum (that is, based on the whole length simulation) is kept in `$meanExp`. Note that the organization differs between the two methods since the first value systematically corresponds to the largest Lyapunov exponent with method 1, whereas it corresponds to the flow direction with method 2:

```
# For method 1 (Wolf et al. 1985)
outLyapFD$Wolf$meanExp
##          [,1]      [,2]      [,3]
## [1,] 0.1584324 0.01690274 -2.881165
#
# For method 2 (Grond et al. 2003)
outLyapFD$Grond$meanExp
##          [,1]      [,2]      [,3]
## [1,] 0.0053202 0.1672354 -2.8784
```

The exponents may thus be reordered to have the Lyapunov spectrum in decreasing order (that is, when at least one of the exponents is positive). The results of the two algorithms are very similar for both methods once reordered: (0.1584; 0.0169; -2.881) with method 1 and (0.1672; 0.0053; -2.878) with method 2.

To have an idea of the robustness of the estimates, the dispersion of averaged values based on the last iterations can be computed and stored in the sublist `$stdExp`:

```

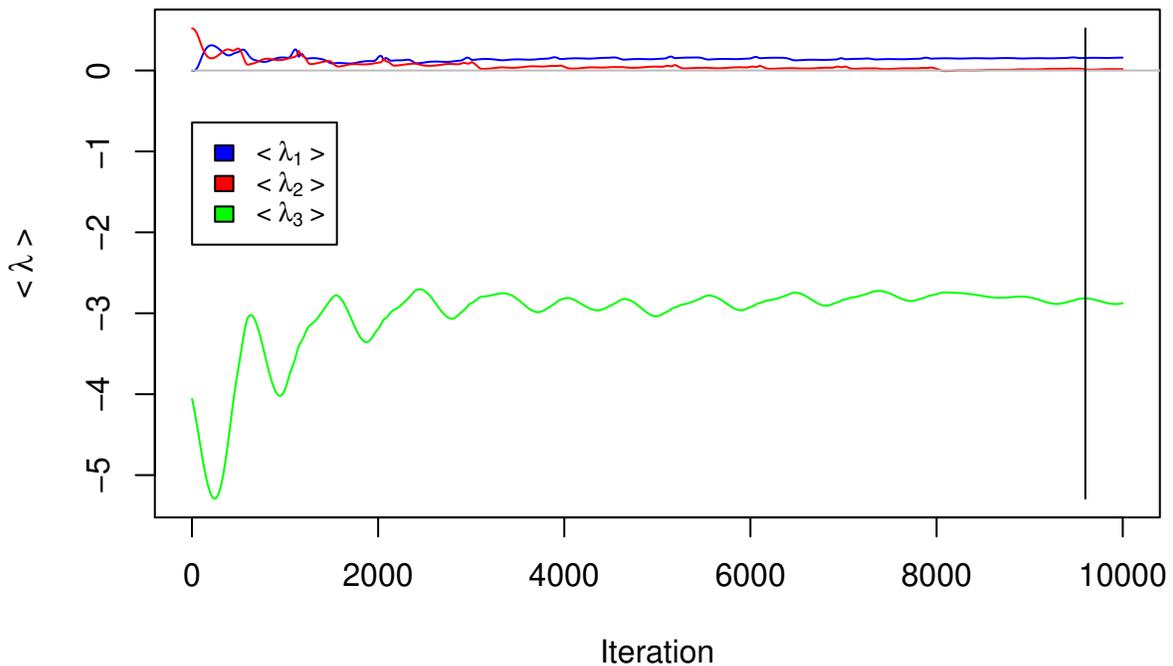
# For method 1 (Wolf et al. 1985)
outLyapFD$Wolf$stdExp
##           [,1]           [,2]           [,3]
## [1,] 0.000265668 7.365932e-05 0.002546517
#
# For method 2 (Grond et al. 2003)
outLyapFD$Grond$stdExp
##           [,1]           [,2]           [,3]
## [1,] 0.0001579498 0.0004809231 0.002542685

```

By default, this dispersion is computed on the last `nIterStats = 50` iterations of the algorithm. However, to have a proper estimate of the error, this number of iterations should be adapted carefully in order to ensure a good representativity over the visited attractor. Indeed, following the evolution of the convergence of the Lyapunov exponents, it is observed that their values will not converge monotonically, but will present oscillations:

```
plotMeanExponents(nVar, outLyapFD$Wolf, nIterStats = 400, xlim = c(1,9999), legend=TRUE)
```

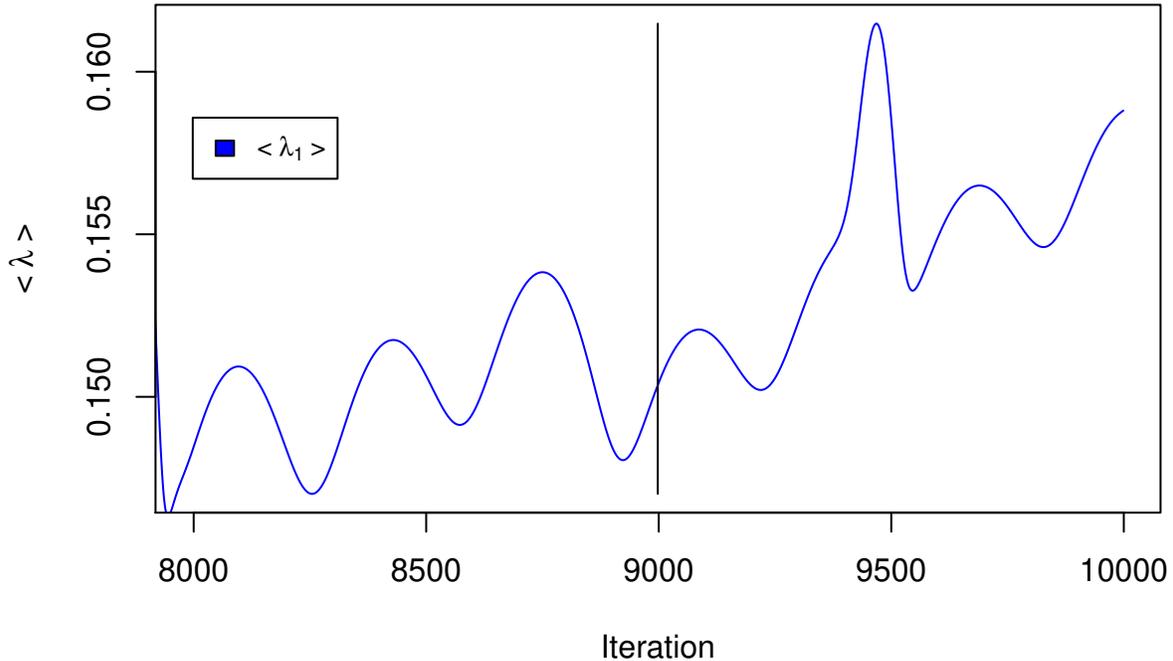
Mean Lyapunov exponents



If we focus on the 2000 last iterations of the largest exponent of the Rössler system (next figure), oscillations remain clearly distinguishable. Their characteristic time relates to the pseudo-period of the Rössler system (indeed, each oscillation through the attractor will generate an oscillation on the averaged value of the local Lyapunov exponents that will also transfer to the global estimates).

```
plotMeanExponents(nVar, outLyapFD$Wolf, nIterStats = 1000, expList = c(TRUE, FALSE, FALSE), legend=TRUE)
```

Mean Lyapunov exponents



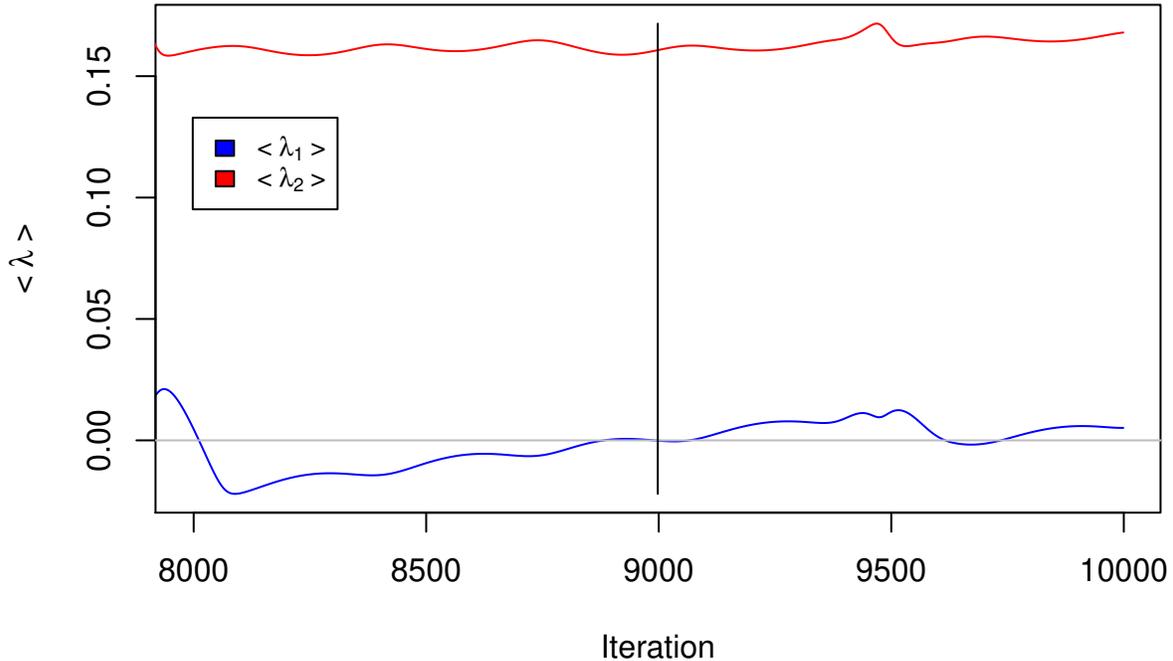
To have an information about the robustness of the estimates, an error bar should also be estimated. This can be done considering the dispersion of the estimated values. This dispersion is automatically assessed on the `nIterStats = 50` last iterations. However, `nIterStats` should be chosen carefully according to the context. When oscillations are easily depicted visually, this parameter should correspond to an integer number of oscillations around the attractor in the phase space, so that its value can be representative of a completed number of loops. The interface provided with the package can be very useful to help fixing this algorithmic parameter. Note that for phase-noncoherent chaotic regimes (which is the case for the parameterization of the chosen example), oscillations may vary both in pseudo-period and amplitude. In the present case, for `nIterStats = 1000`, more or less three oscillations are observed between iterations 9000 and 10000.

As discussed upper in the text, the method introduced by *Grond et al.* (2003) enables to distinguish the direction of the flow from the other directions. Thanks to this, this method allows a finer estimate of the whole Lyapunov spectrum by avoiding any confusion between this exponent from the other exponents, and thus clearly distinguishing also positive from negative exponents.

As already mentioned upper in the text, the mean Lyapunov exponent corresponding to this special direction should equal zero in average. However, it may oscillate - even along the flow direction - since the distance between two points successively positioned along the same trajectory will also vary due to successive accelerations and slowdowns. Despite these variations, the averaged value should be identical since these two points will strictly follow the same trajectory.

```
plotMeanExponents(nVar, outLyapFD$Grond, nIterStats = 1000, expList = c(TRUE, TRUE, FALSE), legend=TRUE)
```

Mean Lyapunov exponents



Considering this observation, the Lyapunov exponents spectrum should be closer to optimal when the direction corresponding to the flow equals zero, since, under these conditions, one of the exponents will effectively have the proper averaged value. Note that such a choice will not be applicable with method 1 since the direction of the flow cannot be systematically distinguishable with this method. Here also, the interface made available with the FDLyapu package may be very useful for a visual choice of the algorithm parameterization.

Based on the run presented in the previous plots, the following Lyapunov spectrum were obtained: $(0.1584 \pm 3.10^{-4}; 0.0169 \pm 1.10^{-4}; -2.881 \pm 3.10^{-3})$ with method 1, and $(0.1672 \pm 4.10^{-4}; 0.0053 \pm 2.10^{-4}; -2.878 \pm 3.10^{-3})$ with method 2.

The Kaplan-Yorke dimension

Similarly, the Kaplan-Yorke dimension is also stored in the outputs. The local dimension is available in `$DkyLoc`, the averaged values (reestimated at each iteration) are in `$Dky`. The last estimate of the dimension is kept in `$meanDky`

```
# For method 1 (Wolf et al. 1985)
outLyapFD$Wolf$meanDky
#
# For method 2 (Grond et al. 2003)
outLyapFD$Grond$meanDky
```

with a standard deviation also kept in `$stdDky`:

```
# For method 1 (Wolf et al. 1985)
outLyapFD$Wolf$stdDky
#
# For method 2 (Grond et al. 2003)
outLyapFD$Grond$stdDky
```

Results of the two algorithms are thus similar: (2.0608 ± 2.10^{-4}) with *Wolf et al.* (1985) and (2.0599 ± 2.10^{-4})

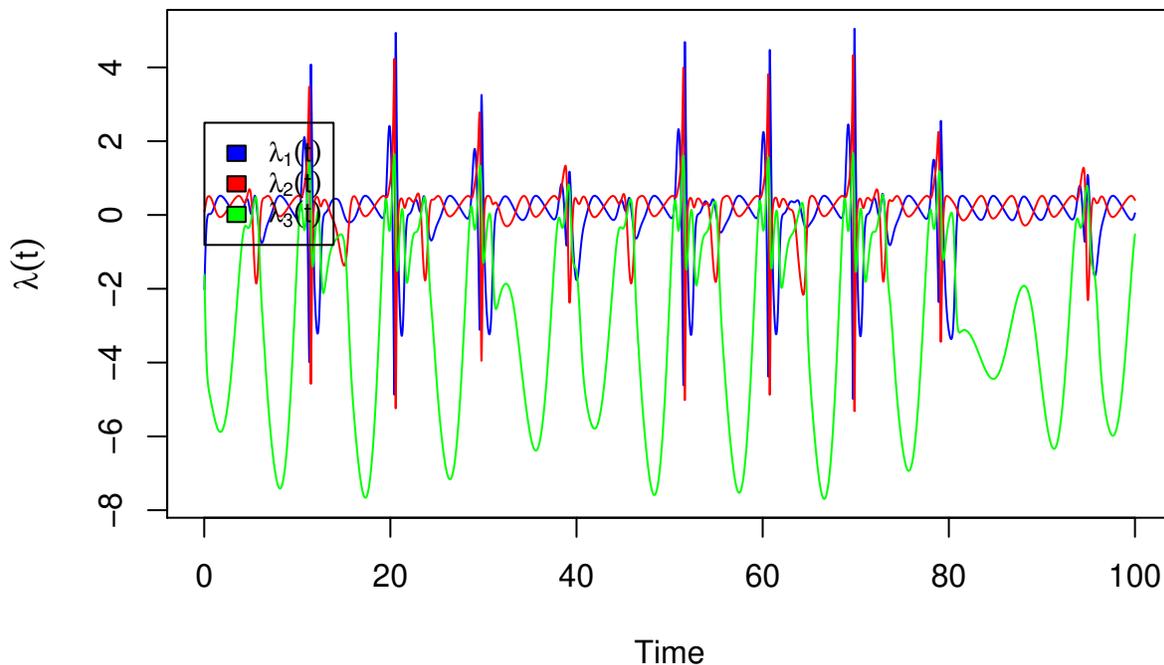
with *Grond et al.* (2003), but the difference between the two estimates is larger than the estimated standard deviation (even at two σ). The ability of the algorithm developed by *Grond et al.* to distinguish the flow direction from the other direction makes it more reliable. Moreover, it enables to converge quicker. The results obtained with the second method can thus be preferred here. A higher precision would be obtained by considering a longer simulation length ($t_{Fin} > 10$).

The local Lyapunov exponents

One specific interest of the method 2 developed by *Grond et al.* (2003) comes from its local validity that allows the analysis of local Lyapunov exponents along the trajectory. The local Lyapunov exponents are also stored in the outputs, they can be plotted for analysis.

```
plotLocalExponents(nVar, outLyapFD$Grond, legend=TRUE)
```

Local Lyapunov exponents



Their time evolution shows the complexity of their behavior.

Interface

An interface is provided with the package. It can be launched either directly running the corresponding `ui.R` application from the `Rstudio` window (“Run App” button) or using the following command

```
shiny::runApp('./inst/shiny-examples/FDLyapu-app')
```

Once launched, the equations of the studied system must be loaded using the interface (“Choose File”) from where several examples are provided with the package:

- The Lorenz-1963 three-dimensional system (`ex3D_Lorenz_1963.R`),
- The hyperchaotic Rössler-1979 system (`ex4D_Rossler_1979.R`),
- The four-dimensional Ebola model obtained from observational data using the `GPOM` package by *Mangiarotti et al.* in 2016 (`ex4D_EbolaModel_2016.R`),

- The five-dimensional hyperchaotic system introduced by *Vaidyanathan et al.* in 2015 (`ex5D_Dynamo_2015.R`),
- And the nine-dimensional hyperchaotic system introduced by *Reiterer et al.* in 1998 (`ex9D_RayleighBenard_1998.R`).

These can all be found in the following folder `.\Examples` provided with the package.

The interface is separated in five tabs: two tabs dedicated to computation (one for each method), two others for the visualization of the local estimates in the phase space (one for each method also), and one more for displaying the equations of the studied system. A detailed description of these tabs is provided hereafter.

⌘ Tabs ‘Wolf’ and ‘Grond’

Among the algorithmic parameters, four ones are directly accessible from the interface: the integration duration `tFin`, the time step `dt`, the plot refreshing frequency (an interface parameter not required to run the algorithm), and the window width to be used for estimating the statistics `nIterStats` (other parameters must be provided inside the loaded file). If these four parameters are provided with the loaded file, the interface will take the corresponding values in the interactive input boxes when loading the file. If not, default values will be used. It will remain possible to modify these values at any time, that is before and after having started the computation process.¹⁰

The ‘Start’ button must be clicked to start the computation (note that this function becomes accessible only once an input file was loaded). The computation can be stopped using the ‘Stop’ button, it can also be reinitialized with the ‘Reset’ button. Once stopped, the algorithm can be restarted using the ‘Start’ button. After the computation has reached the end, the computation can be extended by updating the increasing time and clicking again the ‘Start’ button.

Two plot windows are made available in the tab, either for the Lyapunov exponents spectrum, or for the Kaplan-Yorke dimension. This choice can be done by choosing either ‘Lyapunov exponents’ or ‘Local Dky’ in the control panel (radio button).

A figure of three plots will appear after a period of time. The time series of the local Lyapunov exponents is presented on the first panel with one colour for each exponent. The legend can be obtained by ticking the corresponding box on the upper part of the figure. It is also possible to select which exponents should or should not be plotted on the Figure.

The mean Lyapunov exponents are reestimated at each iteration and stored. The obtained time series are presented on the middle panel. These are expected to progressively converge to an optimal value.

To have an estimate of the precision of the mean Lyapunov exponents, the dispersion of the last estimates is assessed based on the `nIterStats` last iterations. The number of iterations may be chosen in order to have an integer number of completed cycles and thus a more representative sampling of the (expected) attractor.

For all the plots, the X and Y scales are automatically updated in order to provide with a continuous monitoring of the algorithmic estimation. To have an expertise on specific parts of the plots, the automatic axes can be deactivated and the focus chosen manually.

The last estimates are provided in ‘Statistics’ (bottom left), that is, successively: the vector of the mean Lyapunov exponents, and the corresponding vector of standard deviation. Kindly remember that the organisation of the organisation of the two vectors differs depending on what method is used:

- For method 1 (Wolf et al. 1985), the exponents are provided from the largest to the smallest value.
- For method 2 (Grond et al. 2003), the same organisation will be used except for the first exponent which systematically corresponds to the flow direction (which should thus progressively converge to zero with possible oscillations around it).

The number of iterations `nIterStats`, based on which the standard deviation is estimated, is also recalled (see: Window min/max iterations).

¹⁰Note that any interaction performed through the interface while the computation is in process will be received at the next update and the corresponding order and will then become active one more step later.

⌘ Tabs ‘3DWolf’ and ‘3DGrond’ (Visualization in the phase space)

This tab aims to visualize the local Lyapunov exponents in three-dimensional projections of the phase space (it can also be applied to the Kaplan-Yorke dimension). The `rgl` package is used for this purpose. To be applied, it is necessary to define the three axes of the 3D projection (by default, these axes will correspond to the three first system variables) and the exponents to be plotted (or the Dky dimension). Once displayed, the projected phase space can be rotated and magnified with the mouse. The colour palette can also be chosen manually in order to improve the readability of the figure.

The analysis can be applied to the results obtained with the two methods (one tab is made available for each). It should be kept in mind when analysing the results that the organisation of the exponents differs in the two cases: for method one, Lyapunov exponents are organized from the largest to the smallest value, whereas for method two, the first value systematically correspond to the direction of the flow.

⌘ Tab ‘Equations’ (for displaying of the equations)

The equations of the studied system can be visualised in the last tab. This information is important for whom would need to check what equations are analyzed by the algorithm.

Two options are provided. The first option aims to chose the variables names. The other one aims to define the number of digits to be edited in addition to a minimum number of digits. Note that this option is purely qualitative, its aim is just to make the equations structure more readable.

Application cases

The following results were obtained applying the algorithms to the examples upper mentioned.

The three-dimensional Lorenz (1963) system

The Lorenz system was introduced by Edouard N. Lorenz in 1963¹¹ by an extreme simplification of the equations of convection. The system reads:

```
## dx/dt = 10 y -10 x
##
## dy/dt = -y + 28 x -x z
##
## dz/dt = -2.667 z + x y
```

The first hyperchaotic system

The first hyperchaotic system was introduced by Otto E. Rössler in 1979.¹²

```
## dx/dt = -z -y
##
## dy/dt = w + 0.25 y + x
##
## dz/dt = 3 + x z
##
## dw/dt = 0.05 w -0.5 z
```

This system is four-dimensional and has a single nonlinearity xz in the third equation. The algorithms can then be used to estimate the Lyapunov exponent spectrum:

¹¹Edward N. Lorenz, Deterministic nonperiodic flow, *J. Atmos. Sci.*, *vol. **20**(2), 130-141, 1963.

¹²O. E. Rössler, An equation for hyperchaos, *Physics Letters A*, **71**, 155-157, 1979.

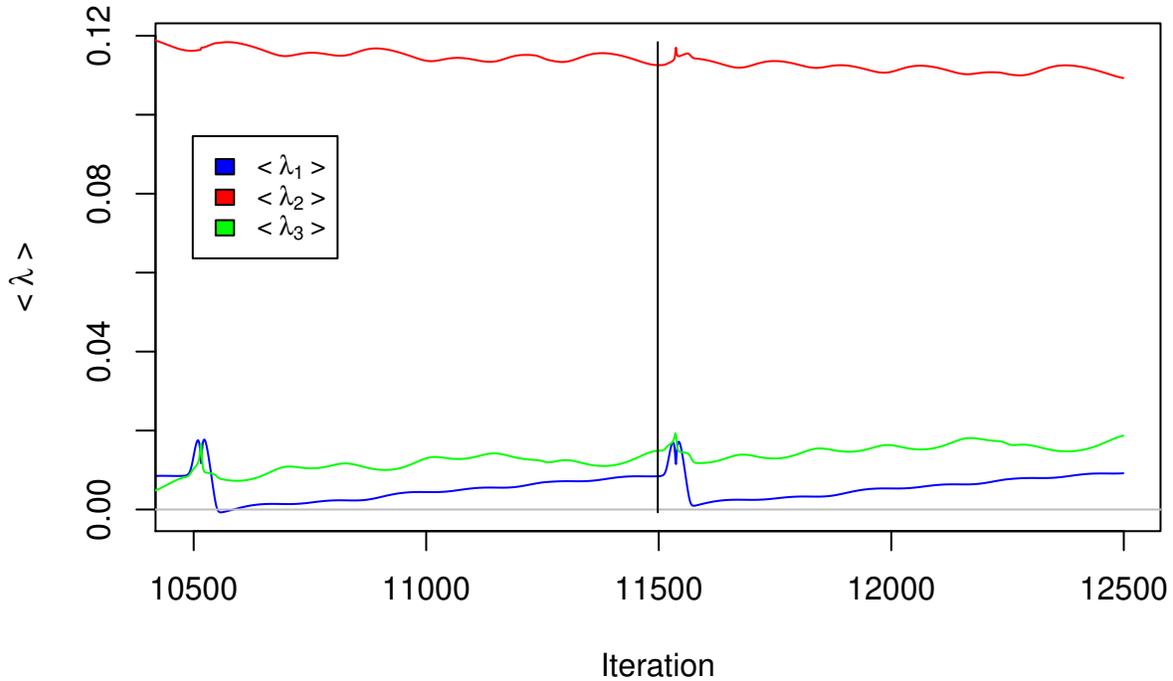
```

# Prepare the output file
outLyapFD <- NULL
# Method 2
outLyapFD$Grond <- lyapFDGrond(outLyapFD$Grond, nVar= 4, dMax = 2, coeffF = K,
                               tDeb = 0, dt = timeStep, tFin = 250, yDeb = c(-10,-6,0,10))

# Plot the results
plotMeanExponents(nVar, outLyapFD$Grond, nIterStats = 1000, expList = c(TRUE, TRUE, TRUE, FALSE), legend

```

Mean Lyapunov exponents



The exponent corresponding to the flow direction is easy to identify when using method 2 (Grond et al.). The exponent corresponding to it is plotted in blue. The subsequent exponents (in red and green respectively) both converge to a positive value.

```

outLyapFD$Grond$meanExp
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.009123498 0.1100914 0.01765182 -20.27899
outLyapFD$Grond$stdExp
##           [,1]      [,2]      [,3]      [,4]
## [1,] 1.457938e-05 0.0005196863 0.0006972188 0.00765801

```

Considering the two largest exponents (respectively in second and third positions) we get $(0.11 \pm 5.2^{-4}; 0.0176 \pm 6.9^{-4})$, and the D_{KY} dimension can also be deduced from it.

```

outLyapFD$Grond$meanDky
##           [,1]
## [1,] 2.00132
outLyapFD$Grond$stdDky
##           [,1]
## [1,] 3.483726e-05

```

We get $D_{KY} = 3.00675 \pm 1.05^{-5}$.

The hyperchaotic character of this system is thus effectively retrieved.

It should be noted however that mean value of the exponent corresponding to the flow direction is relatively far from zero (0.009 ± 1.46^{-5}). Convergence has probably not been reached yet and it would be preferable to consider a longer run to get more precise estimates.

Application to models obtained by the global modelling technique

- The 3D model for the coupled epizootic-epidemic of plague in Bombay (1896-1911)

An epidemic of bubonic plague broke out in Bombay, now Mumbai (India) in 1896. The bacillus *Yersinia Pestis* had been discovered two years before only, and its mode of propagation was still unknown at this time. Numerous investigations were carried out by an Advisory Committee appointed by the *Secretary of State for India*, the *Royal Society* and the *Lister Institute* in 1905.¹³ This committee set up in Bombay and started to explore the plague disease in all possible directions. A considerable quantity of data of remarkable quality was gathered and an important part of it was published in the *Journal of Hygiene* since 1906.¹⁴ Three time series were extracted from this data set: the number of human death (hereafter noted x), the number of contaminated brown rats (y), and the number of contaminated black rats (z). The global modelling technique was applied to this data set¹⁵ from which five global models were obtained. The five models were built on the same algebraic structure given by the following model:

```
# Load the model
data(Plague)
# Model 0 (10-term)
# tuning
KLO <- Plague$models$model193
KLO[7,1] <- KLO[7,1]*0.598
visuEq(nVar = 3, dMax = 2, KLO, approx = 2, substit = 1)
## dx/dt = -0.0936 z^2 + 0.04307 y z -7.21646 x
##
## dy/dt = 0.01026 y^2 -0.02273 x y
##
## dz/dt = 0.04717 z^2 -3.28359 x -0.04858 x z + 0.0059 x y + 0.00436 x^2

# Model 1 (11-term) directly chaotic
KL1 <- Plague$models$model129
visuEq(nVar = 3, dMax = 2, KL1, approx = 2, substit = 1)
## dx/dt = -0.0936 z^2 + 0.04307 y z -12.06766 x
##
## dy/dt = 0.01026 y^2 -0.02273 x y
##
## dz/dt = 0.03026 z^2 + 1.40991 y -6.23298 x -0.03395 x z + 0.00389 x y + 0.00532 x^2
```

This result was surprising because it was the first model for which an interpretation of all the terms could be proposed, that was directly obtained from observational time series. Earlier studies, based on biological considerations, had proven the possibility to transfer the disease from black to brown rats and from black rats to human, but there was no dynamical proof that such a process could develop a large scale epidemic. This model enabled to bring a strong argument for it. This model also revealed the efficiency of human action to slow down the propagation of the disease. Finally, the model brought strong argument for chaos, that is (1) a deterministic dynamics underlying the epidemic, and (2) a high sensitivity to the initial conditions.

- The 4D model for the epidemic of Ebola Virus Disease in West Africa (2013-2016)

¹³Plague Research Commission. The epidemiological observations made by the commission in Bombay City. *Journal of Hygiene*, **7**, 724–98, 1907.

¹⁴<http://www.ncbi.nlm.nih.gov/pmc/journals/326/>.

¹⁵S. Mangiarotti, “Low dimensional chaotic models for the plague epidemic in Bombay (1896–1911), *Chaos, Solitons, Fractals*, **81**(A), 184–196, 2015.

The Ebola model¹⁶ was obtained from observational data using the GPoM package. In December 2013, an epidemic of Ebola Virus Disease broke out in Guinea and spread out in Liberia and Sierra Leone and then became uncontrollable. Data gathered by the World Health Organization was used to model the epidemic. This data, based on the official publications of the Ministries of Health of the three main countries involved in the epidemic, was made available online by the Centers for Disease Control and Protection,¹⁷ reported as cumulative numbers (starting from the beginning of the epidemic).

Two variables were considered in the analysis: the cases of infections and deaths. Since almost all the cases belong to a large scale area including Guinea, Liberia and Sierra Leone, time series resulting from the addition of these three contributions were used. The following model was obtained using the global modelling technique

```
## dI/dt = 0.0001089 D1 D3 + 1.4135051 D1^2 -0.9815931 I D1
##
## dD1/dt = D2
##
## dD2/dt = D3
##
## dD3/dt = 5791.0763272 + 3.7447206 D3 + 2.25e-05 D3^2 -1921.2718525 D2 -0.1614398 D2^2 + 34650.5
```

with I the number of infections, D_1 , the number of deaths, and D_2 , D_3 its first and second derivatives.

Application to higher-dimensional systems

- A 5-D hyperchaotic dynamo system¹⁸ was derived by adding by adding two state feedback controls to the 3D Rikitake two-disc dynamo system introduced by Tsuneji Rikitake in 1958.¹⁹ The resulting 5D system reads:

```
## dX1/dt = 0.1 X5 -1.1 X4 + X2 X3 -X1
##
## dX2/dt = 0.1 X5 -1.1 X4 -X2 -2 X1 + X1 X3
##
## dX3/dt = 1 -X1 X2
##
## dX4/dt = 0.7 X2
##
## dX5/dt = 0.1 X4 + 0.1 X2 + 0.1 X1
```

- The 9-D model for a Rayleigh-Bénard convection in a square cell. The thermal convection in three-dimensional spatial domain are governed by the Boussinesq-Oberbeck equations. By applying a triple Fourier expansion to this set of equations, Reiterer et al. obtained the following nine-dimensional model²⁰:

```
## dC1/dt = -0.2647 C7 + 0.0588 C4^2 + 1.7647 C3 C5 -C2 C4 -1.8889 C1
##
## dC2/dt = -0.25 C9 + C4 C5 -0.5 C2 -C2 C5 + C1 C4
##
## dC3/dt = 0.2647 C8 -1.8889 C3 + C2 C4 -0.0588 C2^2 -1.7647 C1 C5
##
## dC4/dt = 0.25 C9 -0.5 C4 + C4 C5 -C2 C5 -C2 C3
```

¹⁶see upper.

¹⁷Centers for Disease Control and Prevention (2014), see <http://www.cdc.gov/vhf/ebola/outbreaks/2014-west-africa/previous-case-counts.html> for Ebola (Ebola Virus Disease).

¹⁸S. Vaidyanathan, V.-T. Pham & C.K. Volos, A 5-D hyperchaotic Rikitake dynamo system with hidden attractors, *The European Physical Journal*, **224**, 1575-1592, 2015.

¹⁹T. Rikitake, Oscillations of a system of disk dynamos, *Mathematical Proceedings of the Cambridge Philosophical Society*, **54**(1), 89-105, 1958.

²⁰P. Reiterer, C. Lainscsek, F. Schürer, C. Letellier & J. Maquet, A nine-dimensional Lorenz system to study high-dimensional chaos, *Journal of Physics A*, **31**, 7121-7139, 1998.

```

##
## dC5/dt = -0.2222 C5 -0.5 C4^2 + 0.5 C2^2
##
## dC6/dt = -3.5556 C6 -C4 C9 + C2 C9
##
## dC7/dt = -3.7778 C7 + 2 C5 C8 -C4 C9 -43.3 C1
##
## dC8/dt = -3.7778 C8 -2 C5 C7 + 43.3 C3 + C2 C9
##
## dC9/dt = -C9 + 43.3 C4 + C4 C7 + 2 C4 C6 -43.3 C2 -C2 C8 -2 C2 C6

```

Conclusion

In this tutorial, it is presented how to use the `FDLyapu` package to estimate the Lyapunov exponents spectrum and the Kaplan-Yorke dimension, from ODEs in polynomial form which may be either defined manually, or derived from observational time series using the `GPoM` package.