Integration of Ordinary Differential Equations

- Ordinary Differential Equations (ODE, ODEs)
- Numerical methods for ordinary differential equations
  - Euler method
  - Runge-Kutta methods
    - « most widely known member of the Runge–Kutta family is generally referred to as “RK4”, “classical Runge–Kutta method” or simply as “the Runge–Kutta method »
  - Adaptative stepsize control for Runge-Kutta methods
  - Predictor–corrector method
  - Richardson extrapolation

Applications

- chemical kinetics
- population dynamics, Logistic function (and equation)
  - Pierre François Verhulst model and non-linearity
- bifurcation, period doubling and routes to chaos
- Oscillating chemical reactions : Belousov-Zhabotinsky, Brusselator, Oregonator
- Lotka–Volterra equations (predator–prey equations)
- molecular dynamics, Verlet integration

Lorenz system

> « The Lorenz system is a system of ordinary differential equations (the Lorenz equations, note it is not Lorentz) first studied by Edward Lorenz. It is notable for having chaotic solutions for certain parameter values and initial conditions. In particular, the Lorenz attractor is a set of chaotic solutions of the Lorenz system which, when plotted, resemble a butterfly or figure eight. »

- Lorenz system
- Strange attractor
- Numerical solutions (and codes) :
  - http://www.node99.org/tutorials/ar/ (RK4)
  - http://www.gribblelab.org/compneuro2012/2_Modelling_Dynamical_Systems.html#orgheadline5
  - https://titanlab.org/2010/04/08/lorenz-attractor/
  - http://ms08035.blogspot.be/2012/04/program-to-simulate-lorenz-system-in.html
  - http://jakevdp.github.io/blog/2013/02/16/animating-the-lorentz-system-in-3d/
  - with Ipython notebook
    - http://nbviewer.ipython.org/github/pjpmarques/Modelling-the-World/blob/master/Lorenz%20Attractor.ipynb,
    - http://nbviewer.ipython.org/gist/dpsanders/d417c1ffbb76f13f678c, including a tutorial on ODEs
    - https://ipywidgets.readthedocs.io/en/stable/examples/Lorenz%20Differential%20Equations.html, in the Jupyter Widgets documentation
      - https://github.com/jupyter-widgets/ipywidgets/blob/master/docs/source/exam

ples/Lorenz%20Differential%20Equations.ipynb
    - https://github.com/jupyterlab/jupyterlab-demo/blob/master/notebooks/Lorenz.ipynb
- On active state
    - Synchronized Chaos using Lorenz Attractor
    - Lorenz Attractor

Python libraries

- reference guide for odeint from scipy.integrate module
- API doc and examples
- on Stackoverflow :
    - Using adaptive step sizes with scipy.integrate.ode
    - Drawing phase space trajectories with arrows in matplotlib
    - numerical ODE solving in python
    - What is the difference between scipy.integrate.odeint and scipy.integrate.ode?

Références

- Numerical recipes, The Art of Scientific Computing 3rd Edition, William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, 2007, isbn: 9780521880688
    - http://numerical.recipes/
    - http://www2.units.it/ipl/students_area/imm2/files/Numerical_Recipes.pdf, p 707...
    - http://apps.nrbook.com/empanel/index.html#

From:
https://dvillers.umons.ac.be/wiki/ - **Didier Villers, UMONS - wiki**

Permanent link:
https://dvillers.umons.ac.be/wiki/teaching:methcalchim:numerical_methods_for_ordinary_differential_equations?rev=1542704409

Last update: **2018/11/20 10:00**