Introduction to standard and non-standard Numerical Methods

Dr. Mountaga LAM

AMS : African Mathematic School 2018

May 23, 2018

Solving $\frac{dx}{dt} = f(x)$

Are numerical methods whose to forward a step, only the previous step information is needed, ie step n+1 only depends on the step n.

- Euler's idea

Solving $\frac{dx}{dt} = f(x)$

Are numerical methods whose to forward a step, only the previous step information is needed, ie step n+1 only depends on the step n.

- Euler's idea
- I can't solve the equation because I don't know what dx is. So dt pick a small number $h > 0$ and say that

$$\frac{dx}{dt} \approx \frac{x(t+h) - x(t)}{h}$$

Solving $\frac{dx}{dt} = f(x)$

Are numerical methods whose to forward a step, only the previous step information is needed, ie step n+1 only depends on the step n.

- Euler's idea
- I can't solve the equation because I don't know what dx is. So dt pick a small number $h > 0$ and say that

$$\frac{dx}{dt} \approx \frac{x(t+h) - x(t)}{h}$$

- The differential equation then becomes

$$\frac{x(t+h) - x(t)}{h} \approx f(x)$$

Solving $\frac{dx}{dt} = f(x)$

Are numerical methods whose to forward a step, only the previous step information is needed, ie step n+1 only depends on the step n.

- Euler's idea
- I can't solve the equation because I don't know what dx is. So dt pick a small number $h > 0$ and say that

$$\frac{dx}{dt} \approx \frac{x(t+h) - x(t)}{h}$$

- The differential equation then becomes

$$\frac{x(t+h) - x(t)}{h} \approx f(x)$$

- If you know $x(t)$ and h then you can solve this equation for $x(t+h)$.

Solving $\frac{dx}{dt} = f(x)$

- $$\frac{x(t+h) - x(t)}{h} \approx f(x)$$

  has solution

  $$x(t+h) \approx x(t) + h \cdot f(x)$$

Solving $\frac{dx}{dt} = f(x)$

- $$\frac{x(t+h) - x(t)}{h} \approx f(x)$$

  has solution

  $$x(t+h) \approx x(t) + h \cdot f(x)$$

- Exemple ($t = 0$): If we know $x(0)$, then this equation allows us to compute $x(0+h) = x(h)$.
- Exemple ($t = h$): If we know $x(h)$, then this equation allows us to compute $x(h+h) = x(2h)$.
- and than $x(2h+h) = x(3h)$, $x(3h+h) = x(4h)$...

Solving $\frac{dx}{dt} = f(x)$

- Pick small number of h and compute

$$
\begin{array}{ccc}
x(h) & = & x(0) + h \cdot f(0) \\
 & \searrow & | \\
x(2h) & = & x(h) + h \cdot f(h) \\
 & \searrow & | \\
x(3h) & = & x(2h) + h \cdot f(2h) \\
 & \searrow & | \\
x(4h) & = & x(3h) + h \cdot f(3h) \\
 & \vdots &
\end{array}
$$

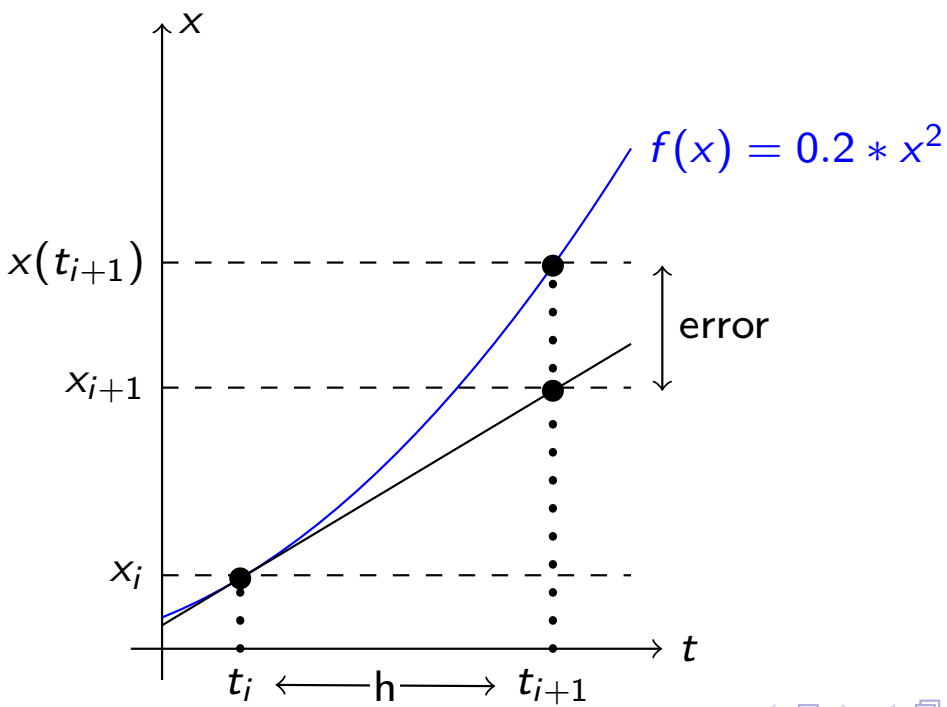
How did Euler do this?
R: By hand (main...), if yes ! How many times ?
How do we do this in the 21st century?
With computer

- Idea is that

$$\text{New value} = \text{Old value} + \text{step size*slope}$$

$$x_{n+1} = x_n + h_n f(t_n, x_n)$$

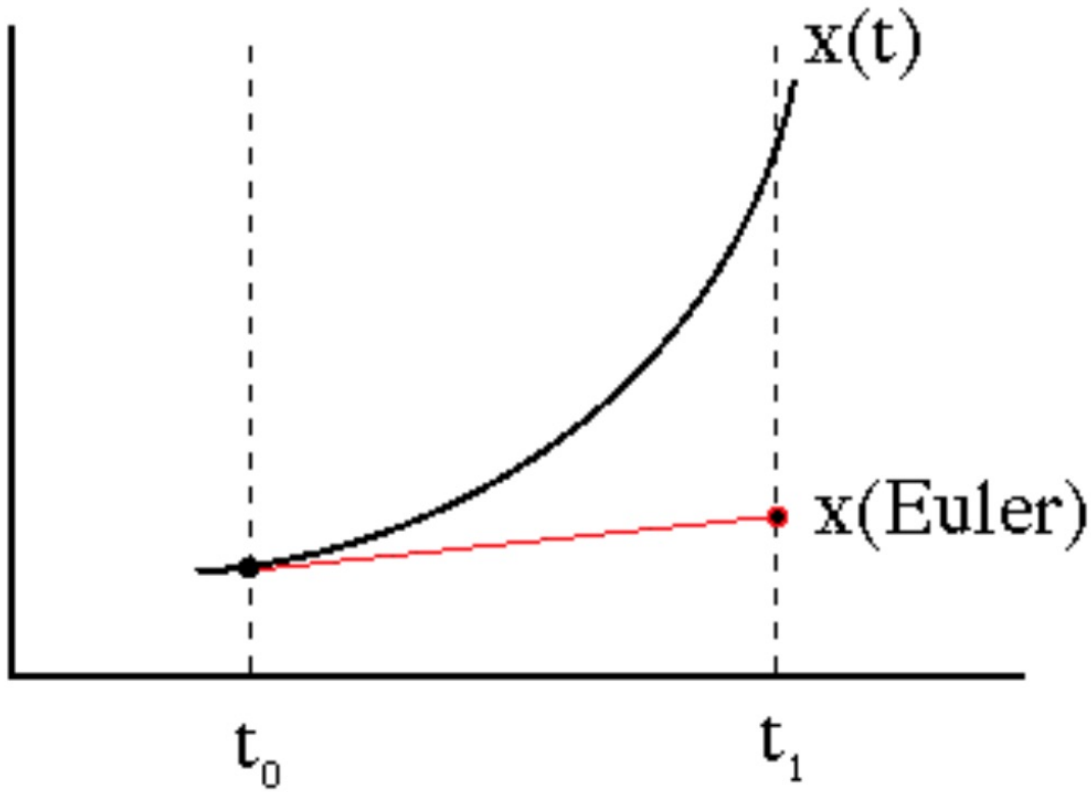
- Slope is generally a function of t, hence $x(t)$
- Different methods differ in how to estimate ϕ



Euler's Method (RK method with order 1)

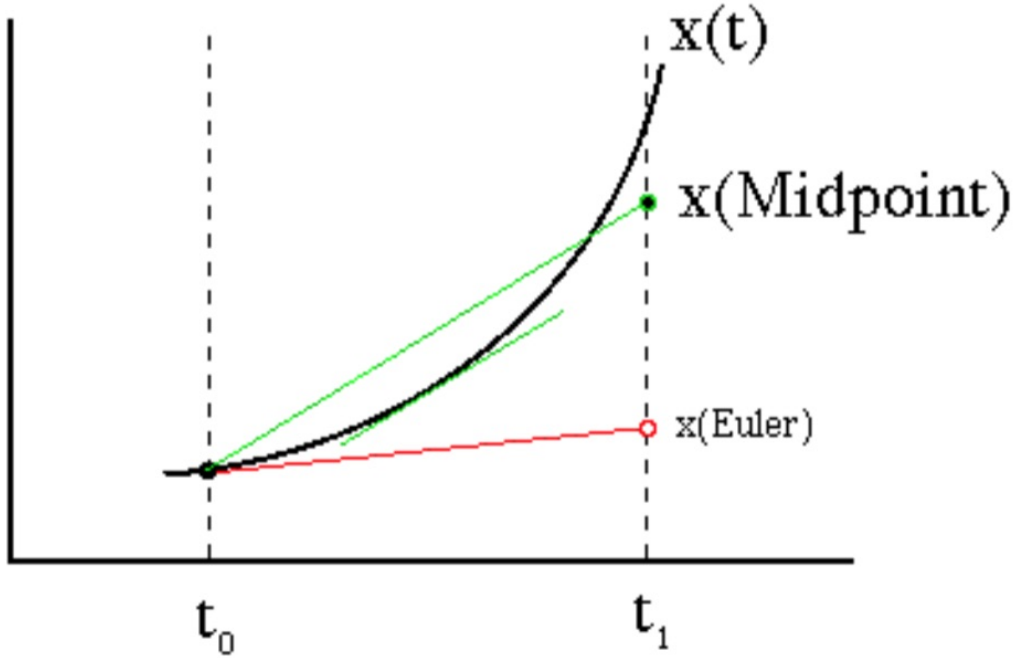
A one-step method expresses x_{n+1} in terms of the previous value x_n.
The simplest example of a one-step method for the numerical solution of the initial value problem (IVP) is Euler's method.
Given that $x(t_0) = x_0$
We define

$$x_{n+1} = x_n + h_n f(t_n, x_n)$$



The midpoint rule (RK method with order two)

$$x_{n+1} = x_n + hf(t_n + \frac{h}{2}, x_n + \frac{h}{2} f(x_n, t_n))$$



Second-Order Runge-Kutta Methods

Let

$$x^{'}(t) = f(t, x(t)).$$

$$x(t+h) = x(t) + hx^{'}(t) + \frac{h^2}{2!}x^{''}(t) + \mathcal{O}(h^3)$$

where

$$x^{''}(t) = f_t(t,x) + f_x(t,x)x^{'}(t) = f_t(t,x) + f_x(t,x)f(t,x)$$

with Jacobian f_x

$$x(t+h) = x(t) + hf(t,x) + \frac{h^2}{2!}\Big[f_t(t,x) + f_x(t,x)f(t,x)\Big] + \mathcal{O}(h^3)$$

Second-Order Runge-Kutta Methods

$$x(t+h) = x(t) + \frac{h}{2} f(t,x) + \frac{h}{2}\Big[f_t(t,x) + h f_t(t,x) + h f_x(t,x) f(t,x)\Big] + \mathcal{O}(h^3)$$

Recalling the multivariate Taylor expansion

$$f(t+h, x+k) = f(t,x) + h f_t(t,x) + f_x(t,x)k + \cdots$$

Then

$$x(t+h) = x(t) + \frac{h}{2} f(t,x) + \frac{h}{2}\Big[f(t+h, x+hf(t,x))\Big] + \mathcal{O}(h^3)$$

we get the numerical method :

$$x_{n+1} = x_n + h\Big(\frac{1}{2}k_1 + \frac{1}{2}k_2\Big)$$

with

$$k_1 = f(t_n, x_n) \quad \text{and} \quad k_2 = f(t_n + h, x_n + hk_1)$$

Fourth-Order Runge-Kutta Methods

The classical method is given by

$$x_{n+1} = x_n + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$

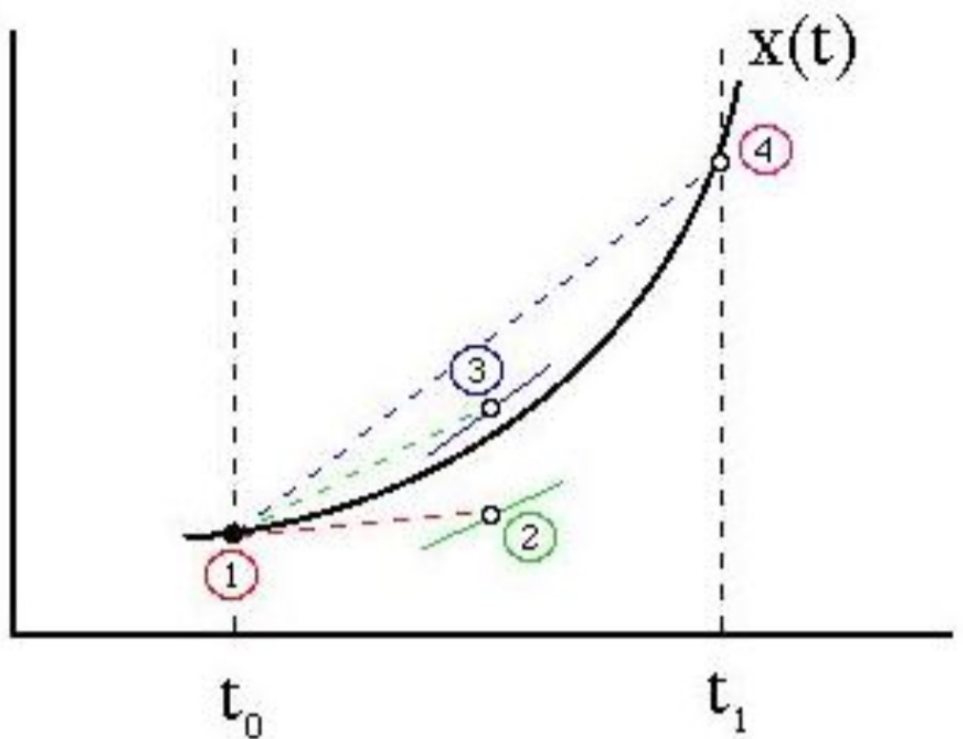
with

k_1 is the increment based on the slope at the beginning of the interval, using x (Euler's method);

k_2 is the increment based on the slope at the midpoint of the interval, using x and k_1

k_3 is again the increment based on the **slope** at the midpoint, but now using x and k_2

k_4 is the increment based on the slope at the end of the interval, using x and k_3



Butcher tableau

The family of explicit Runge-Kutta methods is a generalization of the RK4 method mentioned above. It is given by

$$x_{n+1} = x_n + h \sum_{i=1}^{s} b_i k_i$$

where

$$\begin{aligned}
k_1 &= f(t_n + 0h, x_n), \\
k_2 &= f(t_n + c_2 h, x_n + h(a_{21} k_1)), \\
k_3 &= f(t_n + c_3 h, x_n + h(a_{31} k_1 + a_{32} k_2)), \\
k_4 &= f(t_n + c_4 h, x_n + h(a_{41} k_1 + a_{42} k_2 + a_{43} k_3)), \\
&\vdots \\
k_s &= f(t_n + c_s h, x_n + h(a_{s1} k_1 + a_{s1} k_2 + \cdots + a_{s,s-1} k_{s-1})),
\end{aligned}$$

And the final step to combine these intermediate steps like this:

$$x_{n+1} = x_n + b_1 k_1 + b_2 k_2 + b_3 k_3 + b_4 k_4 + \cdots + b_s k_s$$

The Runge-Kutta method is consistent if

$$\sum_{j=1}^{i-1} a_{ij} = c_i, \quad \text{for} \quad i = 2, \cdots, s.$$

Exemple : The explitcit RK4 method falls in this framework

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

It is not difficult to construct s-stage implicit methods which are A-stable.
For example, this can be done by choosing the coefficients c_i and b_i to be the quadrature points and weights respectively in the Gauss quadrature formula for the evaluation of

$$\int_0^1 f(t)dt \approx \sum_{i=1}^{s} b_i f(c_i)$$

The numbers a_{ij} can then be chosen so that the method has order 2s, and is A-stable.

Let us observe that on expanding $x(t_{n+1}) = x(t_n + h)$ into a Taylor series

$$x^{'}(t_n) = f(t_n, x(t_n))$$

we have that :

$$x(t_{n+1}) = x(t_n) + hf(t_n, x(t_n)) + \mathcal{O}(h^2)$$

More generally, a one-step method may be written in the form

$$x(t_{n+1}) = x(t_n) + h\phi(t_n, x_n, h), \quad n = 0, 1, \cdots, N-1, \quad x(t_0) = x_0,$$

where we assume that $\phi : [t_0, t_0 + T] \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a continuous function.
In pratical case, the function $\phi(t, x, h)$ can be define
For example : Euler's Method is given by $\phi(t_n, x_n, h) = f(t_n, x_n)$

Global error

In order to assess the accuracy of the numerical method, we define the **global error**, e_n, by

$$e_n = x(t_n) - x_n$$

The truncation error, T_n, is define by

$$T_n = \frac{x(t_{n+1}) - x(t_n)}{h} - \phi(t_n, x(t_n); h).$$

Convergence

Consider the general one-step method where, in addition to being a continuous function of its arguments, ϕ is assumed to satisfy a Lipschitz condition with respect to its second argument, that is, there exists a positive constant L_ϕ such that, for $0 \leq h \leq h_0$ and for (t, u) and (t, v) in the rectangle

$$D = \{(t, x) : t_0 \leq x \leq t_M, \quad |x - x_0| \leq C\}$$

we have that

$$|\phi(t, u; h) - \phi(t, v; h)| \leq L_\phi |u - v|.$$

Then, assuming that $|x - x_0| \leq C$, $n = 1, 2, \cdots, N$ it follows that

$$|e_n| \leq \frac{T}{L_\phi}\left(e^{L_\phi(t_n - t_0)} - 1\right)$$

where $T = \max_{0 \leq n \leq N-1} |T_n|$

Consistance

Let us apply this general result in order to obtain a bound on the global error in Euler's method. The truncation error for Euler's method is given by

$$T_n = \frac{x(t_{n+1}) - x(t_n)}{h} - f(t_n, x(t_n)) = \frac{x(t_{n+1}) - x(t_n)}{h} - x^{'}(t_n)$$

Taylor series :

$$x(t_{n+1}) = x(t_n) + hx^{'}(t_n) + \frac{h^2}{2!}x^{''}(\zeta_n), \quad t_n < \zeta_n < t_{n+1}.$$

$$T_n = \frac{1}{2}hx^{''}(\zeta_n)$$

$$M = \max_{\zeta \in [t_0, t_M]} |x^{''}|$$

$$|T_n| \leq T = \frac{1}{2}M\left[\frac{e^{L(t_n - t_0)} - 1}{L}\right], n = 0, 1, \cdots, N.$$

Remark : In practice, for such h we shall have $|x(t_n) - x_n| = |e_n| \leq Tol$ for any $n = 0, 1, \cdots, N$. So $h \leq$ (expression $*$ Tol)

Bref definition

- A Scheme is said explicit if $x(t_{i+1})$ can be write as a linear combination of $x(t_i), f(x_k, t_{i_1}), \ldots \forall k$. The scheme is implicite if it others values are necessar.

Bref definition

- A Scheme is said explicit if $x(t_{i+1})$ can be write as a linear combination of $x(t_i), f(x_k, t_{i_1}), \ldots \forall k$. The scheme is implicite if it others values are necessar.
- The scheme is said one-step scheme if we use only two values of times (i.e. t_i and t_{i+1}), otherwise it will be a multi step scheme.

Bref definition

- A Scheme is said explicit if $x(t_{i+1})$ can be write as a linear combination of $x(t_i), f(x_k, t_{i_1}), \ldots \forall k$. The scheme is implicite if it others values are necessar.
- The scheme is said one-step scheme if we use only two values of times (i.e. t_i and t_{i+1}), otherwise it will be a multi step scheme.
- The convergence means that the numerical solution of the numerical scheme tend to the solution of the ordinary differentiale equation.

Bref definition

- A Scheme is said explicit if $x(t_{i+1})$ can be write as a linear combination of $x(t_i), f(x_k, t_{i_1}), \ldots \forall k$. The scheme is implicite if it others values are necessar.
- The scheme is said one-step scheme if we use only two values of times (i.e. t_i and t_{i+1}), otherwise it will be a multi step scheme.
- The convergence means that the numerical solution of the numerical scheme tend to the solution of the ordinary differentiale equation.
- The discretisation scheme $M_{h,\Delta t}$ of operator L is consistant if the function ϕ is smooth enought and

$$\lim_{h,\Delta t \to 0} (M\phi - M_{h,\Delta t}\phi) = 0$$

Bref definition

- A Scheme is said explicit if $x(t_{i+1})$ can be write as a linear combination of $x(t_i), f(x_k, t_{i_1}), \ldots \forall k$. The scheme is implicite if it others values are necessar.
- The scheme is said one-step scheme if we use only two values of times (i.e. t_i and t_{i+1}), otherwise it will be a multi step scheme.
- The convergence means that the numerical solution of the numerical scheme tend to the solution of the ordinary differentiale equation.
- The discretisation scheme $M_{h,\Delta t}$ of operator L is consistant if the function ϕ is smooth enought and

$$\lim_{h,\Delta t \to 0} (M\phi - M_{h,\Delta t}\phi) = 0$$

- A linear scheme consistant is convergent if only if it is stable.

Nonstandard Finite Difference Scheme (NFDS)

Exact schemes : Examples

Let

$$\frac{dx}{dt} = f(t,x) = -\lambda x \quad \text{and} \quad x(t_0) = x_0$$

The general solution is given by :

$$x(t) = x_0 e^{-\lambda(t-t_0)}.$$

then the exact scheme is

$$u_{k+1} = e^{-\lambda h} u_k.$$

NFDS : Examples

Let

$$\frac{dx}{dt} = f(t,x) = -\lambda x \quad \text{and} \quad x(t_0) = x_0$$

$$x_{k+1} = e^{-\lambda h} x_k.$$

Combining the previous exact schemes, the NFDS scheme is given by :

$$x_{k+1} - x_k = (e^{-\lambda h} - 1)x_k = -\lambda \left(\frac{1 - e^{-\lambda h}}{\lambda} \right) x_k, \tag{1}$$

than

$$\frac{x_{k+1} - x_k}{\phi(h)} = -\lambda x_k \qquad \text{avec} \qquad \phi(h) = \frac{1 - e^{-\lambda h}}{\lambda} \tag{2}$$

The Lotka-Volterra system: Euler's Method

$$\begin{cases} \frac{x_{k+1}-x_k}{\phi_1(h)} &= ax_k - bx_{k+1}y_k, \\ \frac{y_{k+1}-y_k}{\phi_2(h)} &= -cy_{k+1} + dx_{k+1}y_k, \end{cases}$$

with

$$\phi_1(h) = -\frac{1-e^{-\lambda h}}{\lambda} \quad \text{and} \quad \phi_2(h) = -\frac{1-e^{-\lambda h}}{\lambda}$$

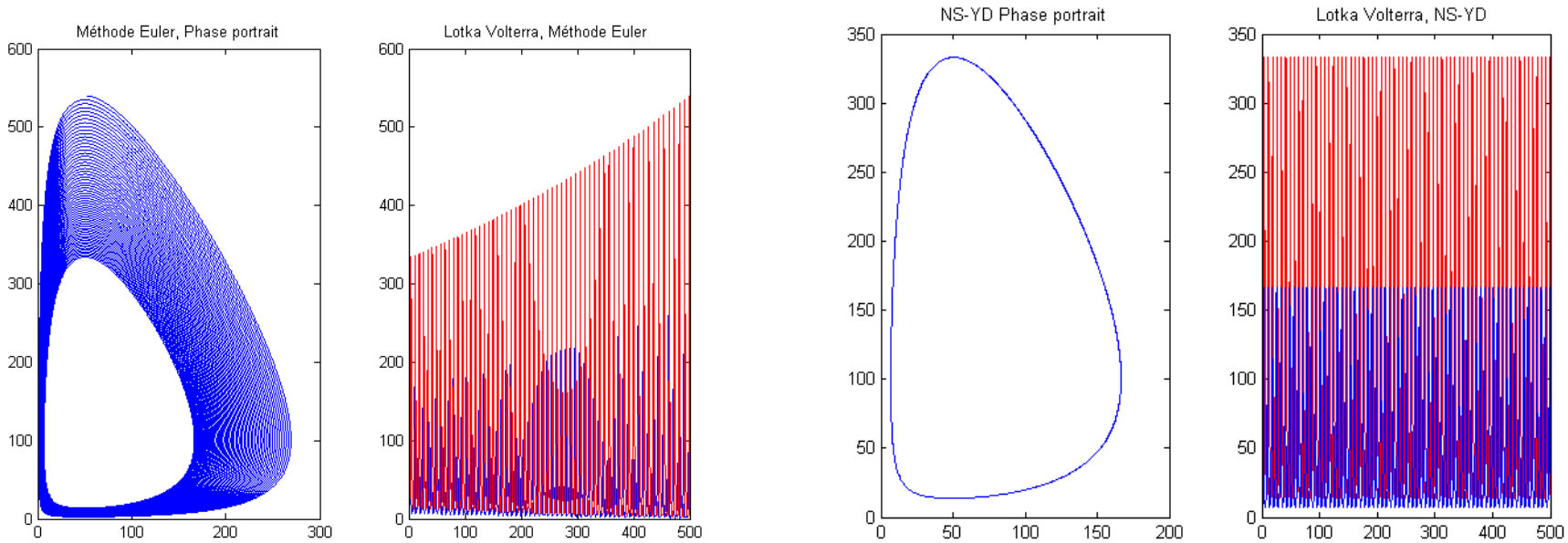


Figure: Phase plane and time history : Euler explicit method and NFDS method of Lotka-Volterra system (a=1;b=0.01;c=1;d=0.02)

The Lotka-Volterra system: Euler's Method

$$\begin{cases} \frac{x_{k+1}-x_k}{\phi_1(h)} &= ax_k - bx_{k+1}y_k, \\ \frac{y_{k+1}-y_k}{\phi_2(h)} &= -cy_{k+1} + dx_{k+1}y_k, \end{cases}$$

with

$$\phi_1(h) = -\frac{1-e^{-\lambda h}}{\lambda} \quad \text{and} \quad \phi_2(h) = -\frac{1-e^{-\lambda h}}{\lambda}$$

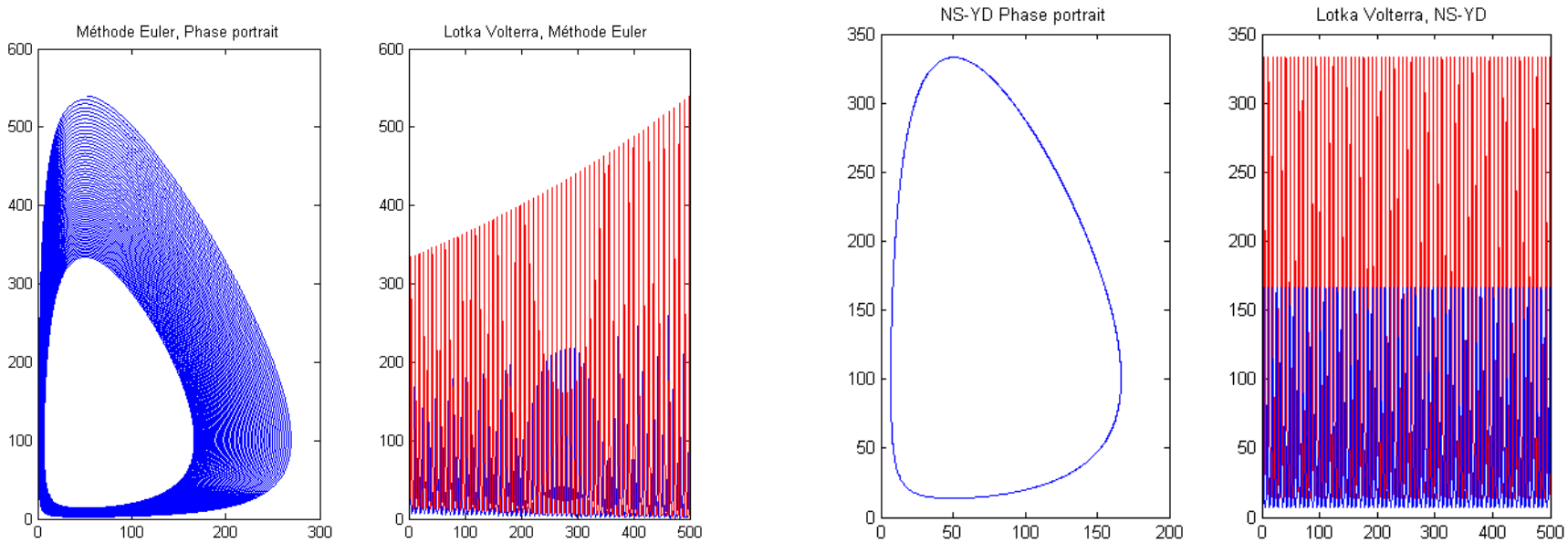


Figure: Phase plane and time history : Euler explicit method and NFDS method of Lotka-Volterra system (a=1;b=0.01;c=1;d=0.02)

Mikens Rules

Pr. Ronald Mickens (Atlanta, USA) used the construction of exact schemes to propose "simple" rules to develop nonstandard finite difference schemes for differential equations and even partial differential equations.

- Rule 1 The orders of the discrete derivatives should be equal to the orders of the corresponding derivatives of the differential equation.

Mikens Rules

Pr. Ronald Mickens (Atlanta, USA) used the construction of exact schemes to propose "simple" rules to develop nonstandard finite difference schemes for differential equations and even partial differential equations.

- Rule 1 The orders of the discrete derivatives should be equal to the orders of the corresponding derivatives of the differential equation.
- Rule 2 Denominator functions for the discrete derivatives must, in general, be expressed in terms of more complicated functions of the step-size than those conventionally used

Mikens Rules

Pr. Ronald Mickens (Atlanta, USA) used the construction of exact schemes to propose "simple" rules to develop nonstandard finite difference schemes for differential equations and even partial differential equations.

- Rule 1 The orders of the discrete derivatives should be equal to the orders of the corresponding derivatives of the differential equation.
- Rule 2 Denominator functions for the discrete derivatives must, in general, be expressed in terms of more complicated functions of the step-size than those conventionally used
- Rule 3 Nonlinear terms should be, in general, be replaced by nonlocal discrete representations.

Mikens Rules

Pr. Ronald Mickens (Atlanta, USA) used the construction of exact schemes to propose "simple" rules to develop nonstandard finite difference schemes for differential equations and even partial differential equations.

- Rule 1 The orders of the discrete derivatives should be equal to the orders of the corresponding derivatives of the differential equation.
- Rule 2 Denominator functions for the discrete derivatives must, in general, be expressed in terms of more complicated functions of the step-size than those conventionally used
- Rule 3 Nonlinear terms should be, in general, be replaced by nonlocal discrete representations.
- Rule 4 Special conditions that hold for the solutions of the differential equations should also hold for the solutions of the finite difference scheme.

Mikens Rules

Pr. Ronald Mickens (Atlanta, USA) used the construction of exact schemes to propose "simple" rules to develop nonstandard finite difference schemes for differential equations and even partial differential equations.

- Rule 1 The orders of the discrete derivatives should be equal to the orders of the corresponding derivatives of the differential equation.
- Rule 2 Denominator functions for the discrete derivatives must, in general, be expressed in terms of more complicated functions of the step-size than those conventionally used
- Rule 3 Nonlinear terms should be, in general, be replaced by nonlocal discrete representations.
- Rule 4 Special conditions that hold for the solutions of the differential equations should also hold for the solutions of the finite difference scheme.
- Rule 5 The scheme should not introduce extraneous or spurious solutions.

Mikens Rules

Pr. Ronald Mickens (Atlanta, USA) used the construction of exact schemes to propose "simple" rules to develop nonstandard finite difference schemes for differential equations and even partial differential equations.

- Rule 1 The orders of the discrete derivatives should be equal to the orders of the corresponding derivatives of the differential equation.
- Rule 2 Denominator functions for the discrete derivatives must, in general, be expressed in terms of more complicated functions of the step-size than those conventionally used
- Rule 3 Nonlinear terms should be, in general, be replaced by nonlocal discrete representations.
- Rule 4 Special conditions that hold for the solutions of the differential equations should also hold for the solutions of the finite difference scheme.
- Rule 5 The scheme should not introduce extraneous or spurious solutions.
- Rule 6 For N differential system, it could be useful to construct nonstandard schemes for subsystems of $M < N$ differential equations and to combine them to obtain a consistant scheme.

Some examples of NFDS of ODE

Équations	NFDS	*Denominators*
$\frac{dx}{dt} = -\lambda x$	$\frac{x_{k+1}-x_k}{\phi(h)} = -\lambda x_k$	$\phi(h) = \frac{1-e^{-\lambda h}}{\lambda}$
$\frac{dx}{dt} = -\lambda x$	$\frac{x_{k+1}-x_k}{\phi(h)} = -\lambda x_{k+1}$	$\phi(h) = \frac{e^{\lambda h}-1}{\lambda}$
$\frac{dx}{dt} = x^2$	$\frac{x_{k+1}-x_k}{\phi(h)} = x_k x_{k+1}$	$\phi(h) = h$
$\frac{dx}{dt} = \lambda_1 x - \lambda_2 x^2$	$\frac{x_{k+1}-x_k}{\phi(h)} = \lambda_1 x_k - \lambda_2 x_k x_{k+1}$	$\phi(h) = \frac{e^{\lambda_1 h}-1}{\lambda_1}$
$\frac{dx}{dt} = \lambda_1 x - \lambda_2 x^2$	$\frac{x_{k+1}-x_k}{\phi(h)} = \lambda_1 x_{k+1} - \lambda_2 x_k x_{k+1}$	$\phi(h) = \frac{1-e^{-\lambda_1 h}}{\lambda_1}$
$\frac{d^2x}{dt^2} + \omega^2 x = 0$	$\frac{x_{k+1}-2x_k+x_{k-1}}{\phi^2(h)} + \omega^2 x_k = 0$	$\phi(h) = \frac{2}{\omega} sin\left(\frac{h\omega}{2}\right)$
$\frac{d^2x}{dt^2} = \lambda \frac{dx}{dt}$	$\frac{x_{k+1}-2x_k+x_{k-1}}{\phi_1(h)} = \lambda \frac{x_{k+1}-x_k}{\phi_2(h)}$	$\phi_1(h) = (\frac{e^{\lambda h}-1}{\lambda})h; \phi_2(h) = h$

Fundamental Rules?

- it is fundamental to follow rules 2 and 3. The others rules then follow...
- to build the best numerical scheme we need a deep theoretical study of the continuous problem is helpful to capture the properties of the solution and the problem

Re-formulation for applying GAS in dynamical systems: Kamgang-Sallet

$$\frac{dx}{dt} = A(x)x + f, \qquad x(0) = x_0,$$

$$\begin{cases} \frac{dS}{dt} &= (1-\pi)\rho - (\tau + \vartheta + \mu)S, \\ \frac{dV}{dt} &= \pi\rho + \tau S - (\mu + \gamma)V, \\ \frac{dI}{dt} &= \vartheta S - (\alpha + \delta + \mu + \kappa)I, \\ \frac{dJ}{dt} &= \delta I - \mu J, \\ \frac{dG}{dt} &= \xi I - \eta G, \\ \frac{dR}{dt} &= \gamma V + \alpha I - \mu R, \end{cases}$$

where

$$\vartheta = \beta_h \frac{I}{N} + \beta_G \frac{G}{G+K}, \quad y = (S, V, R), \quad z = (I, J, G)$$

$$\begin{cases} \frac{dy}{dt} &= A_1(x)(y - y^*) + A_{12}(x)z, \\ \frac{dz}{dt} &= A_2(x)z, \end{cases}$$

Application of GAS with numerical dynamical systems : Kamgang and Sallet

$$\phi(h) \leq min\left\{\frac{1}{\mu+\gamma}, \frac{1}{\alpha+\delta+\mu+\kappa}, \frac{1}{\mu}, \frac{1}{\eta}\right\}.$$

$$\exists Q \text{ such that } Q \geq \max \frac{1}{2}\{-\lambda_i\},\ i = 1, \cdots, 5,$$

chose of $\phi(h)$ is given by

$$\phi(h) = \frac{\varphi(Qh)}{Q} \text{ avec } \varphi(z) = 1 - e^{-z},\ \forall\ z \in \mathbb{R}_+.$$

Implicit scheme

$$\begin{cases} \frac{S^{k+1}-S^k}{\phi(h)} &= (1-\pi)\rho - (\tau + \beta_h \frac{I^k}{N^k} + \beta_G \frac{G^k}{G^k+K} + \mu)S^{k+1}, \\ \frac{V^{k+1}-V^k}{\phi(h)} &= \pi\rho + \tau S^{k+1} - (\mu+\gamma)V^{k+1}, \\ \frac{I^{k+1}-I^k}{\phi(h)} &= (\beta_h \frac{I^k}{N^k} + \beta_G \frac{G^k}{G^k+K})S^{k+1} - (\alpha+\delta+\mu+\kappa)I^{k+1}, \\ \frac{J^{k+1}-J^k}{\phi(h)} &= \delta I^{k+1} - \mu J^{k+1}, \\ \frac{G^{k+1}-G^k}{\phi(h)} &= \xi I^{k+1} - \eta G^{k+1}, \\ \frac{R^{k+1}-R^k}{\phi(h)} &= \gamma V^{k+1} + \alpha I^{k+1} - \mu R^{k+1}, \end{cases}$$

$$\begin{cases} \frac{y^{k+1}-y^k}{\phi(h)} &= A_1(x^k)(y^{k+1}-y^*) + A_{12}(x^k)z^{k+1}, \\ \frac{z^{k+1}-z^k}{\phi(h)} &= A_2(x^k)z^{k+1}, \end{cases}$$

explicit scheme

$$\begin{cases} \frac{S^{k+1}-S^k}{\phi(h)} &= (1-\pi)\rho - (\tau + \beta_h \frac{I^k}{N^k} + \beta_G \frac{G^k}{G^k+K} + \mu)S^{k+1}, \\ \frac{V^{k+1}-V^k}{\phi(h)} &= \pi\rho + \tau S^k - (\mu+\gamma)V^k, \\ \frac{I^{k+1}-I^k}{\phi(h)} &= (\beta_h \frac{I^k}{N^k} + \beta_G \frac{G^k}{G^k+K})S^{k+1} - (\alpha+\delta+\mu+\kappa)I^k, \\ \frac{J^{k+1}-J^k}{\phi(h)} &= \delta I^k - \mu J^k, \\ \frac{G^{k+1}-G^k}{\phi(h)} &= \xi I^k - \eta G^k, \\ \frac{R^{k+1}-R^k}{\phi(h)} &= \gamma V^k + \alpha I^k - \mu R^k, \end{cases}$$

$$\begin{cases} \frac{y^{k+1}-y^k}{\phi(h)} &= A_1(x^k)(y^k - y^*) + A_{12}(x^k)z^k, \\ \frac{z^{k+1}-z^k}{\phi(h)} &= A_2(x^k)z^k, \end{cases}$$

Simulations

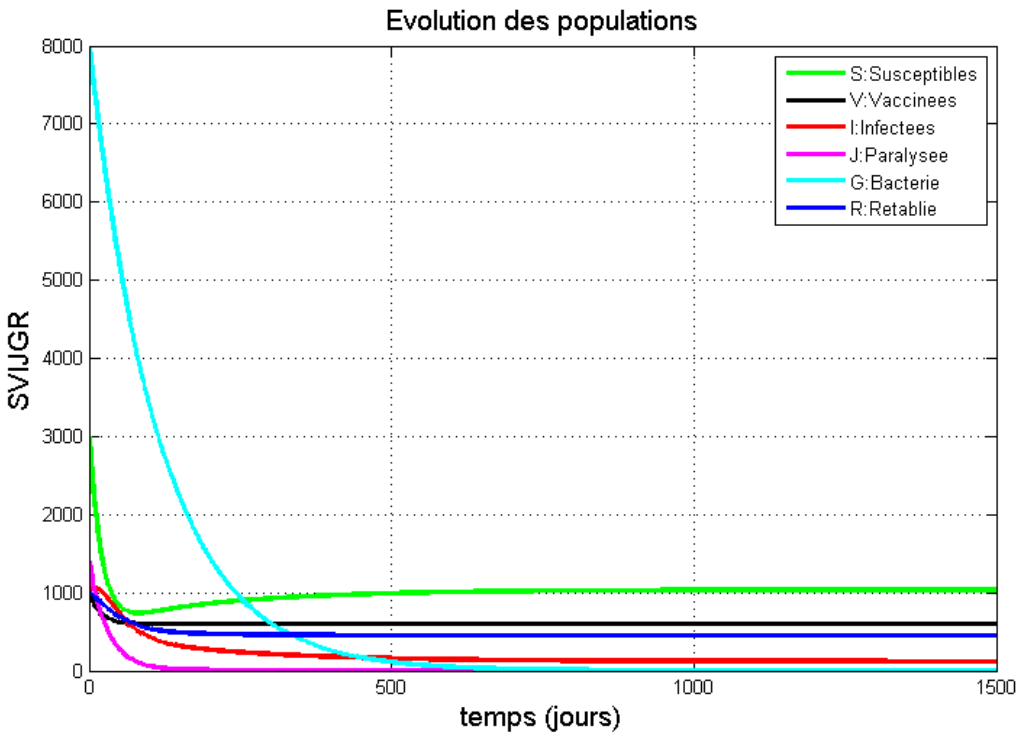


Figure: $\underline{\mathcal{R}_0 > 1.}$

Simulations

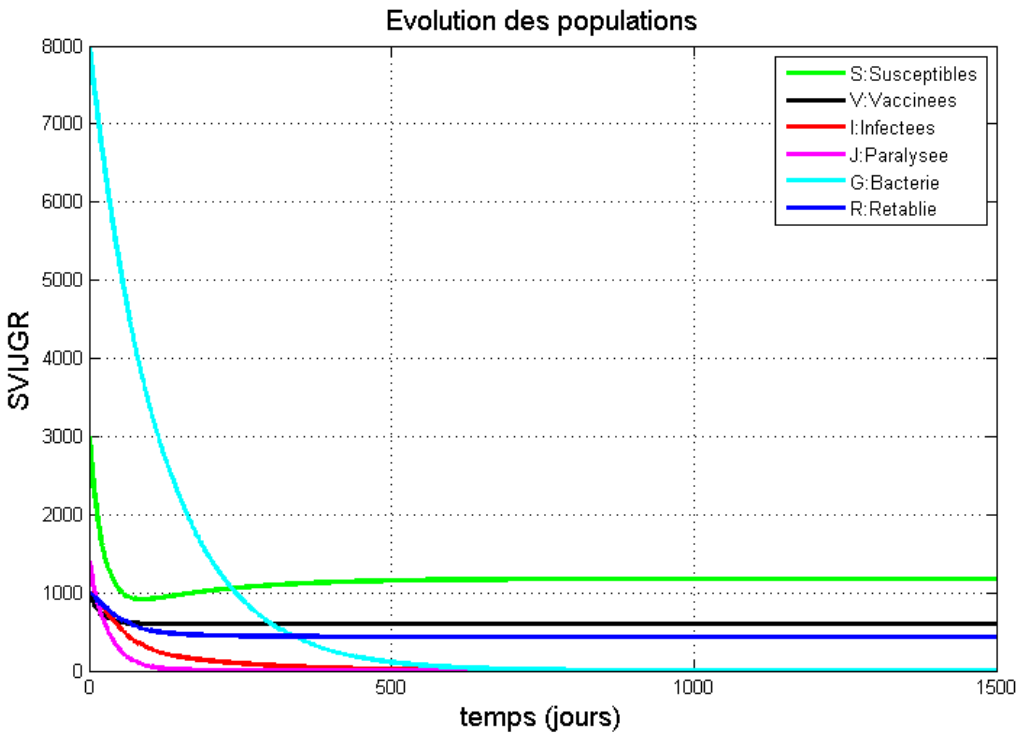


Figure: $\underline{\mathcal{R}_0 < 1.}$

Comparison

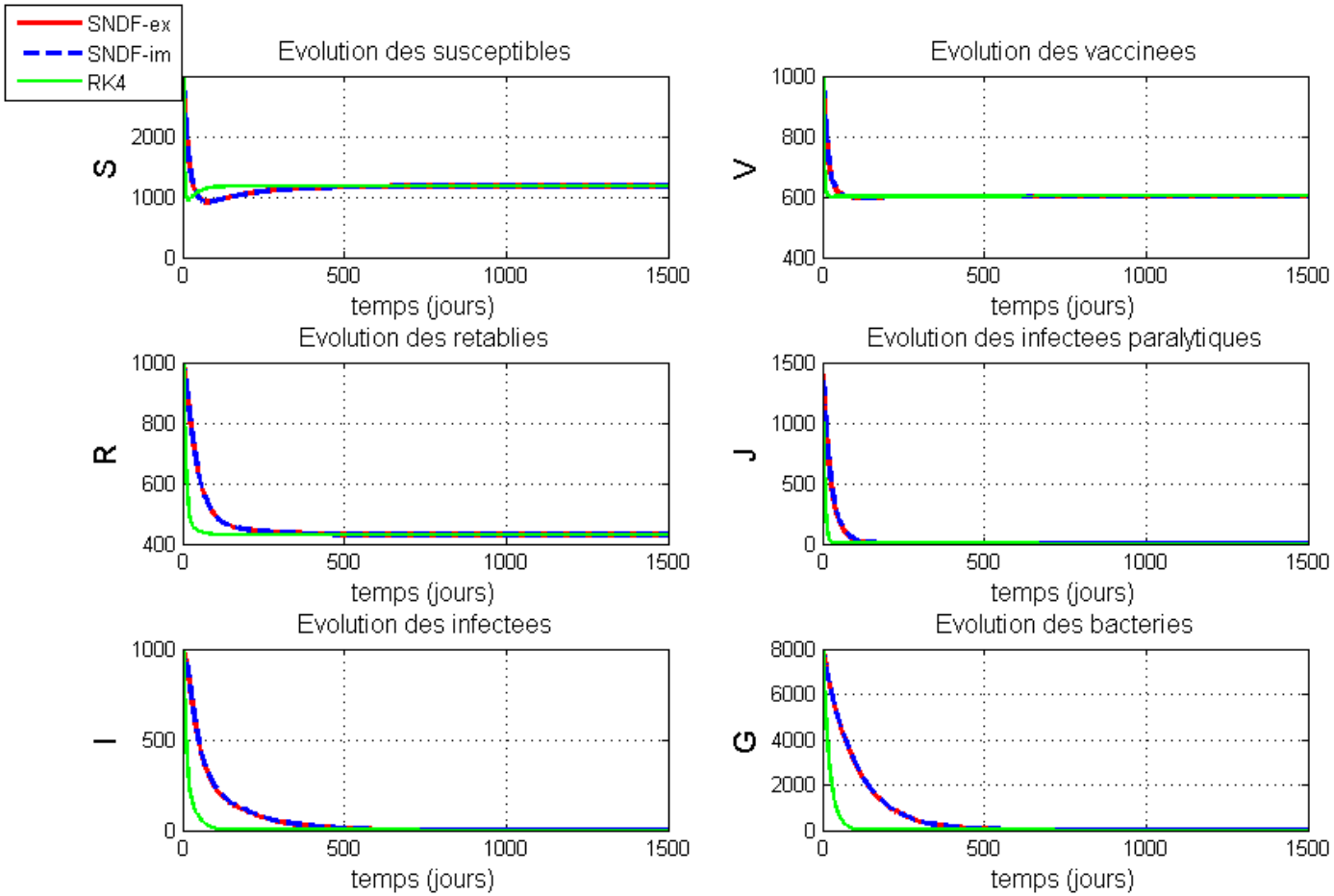


Figure: Comparison between NFDS and RK4

Comparison

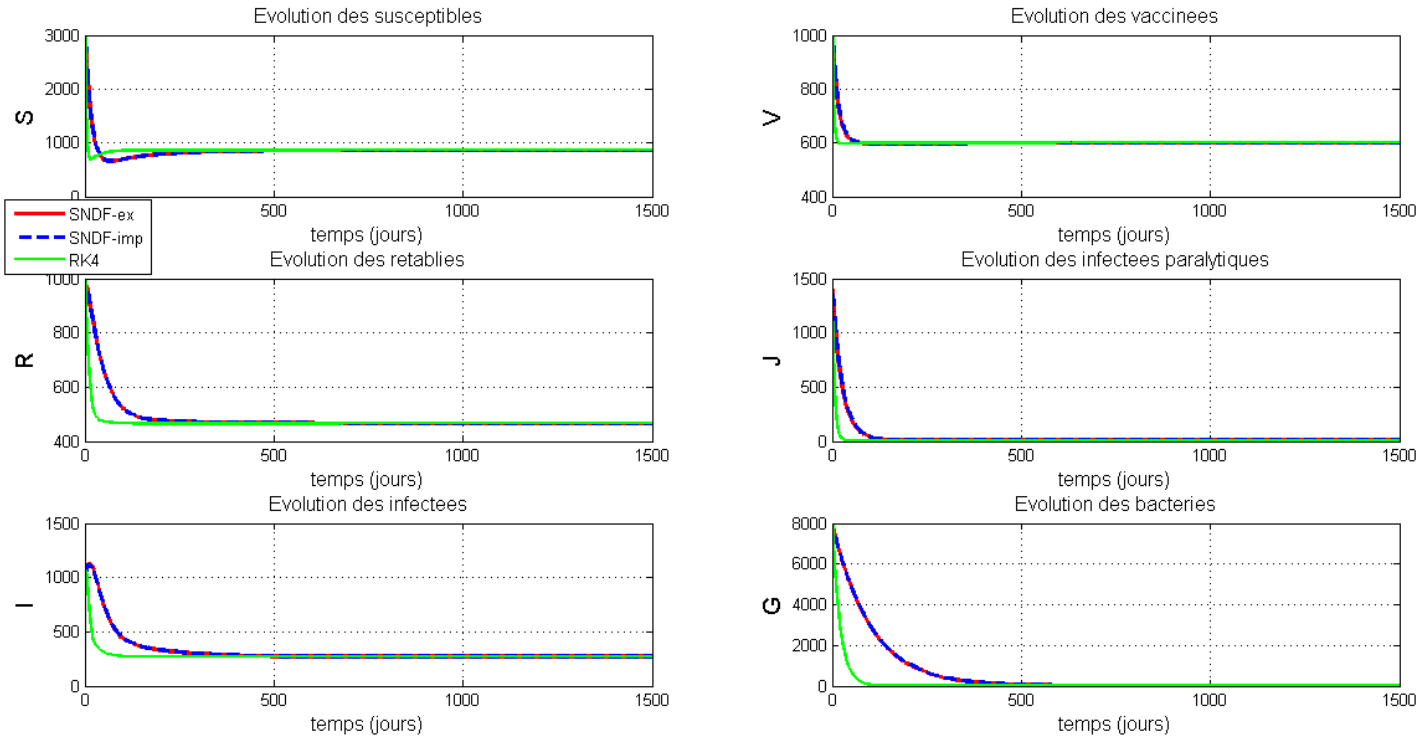


Figure: Comparison between NFDS and RK4