a.In mathematics, a dynamical system is a system in which a function describes the time dependence of a point in a geometrical space. Examples include the mathematical models that describe the swinging of a clock pendulum, the flow of water in a pipe, and the number of fish each springtime in a lake.

At any given time, a dynamical system has a state given by a tuple of real numbers (a vector) that can be represented by a point in an appropriate state space (a geometrical manifold). The evolution rule of the dynamical system is a function that describes what future states follow from the current state. Often the function is deterministic, that is, for a given time interval only one future state follows from the current state. However, some systems are stochastic, in that random events also affect the evolution of the state variables.

b.Dynamical systems are mainly represented by a state that evolves in time. The input as well as the current state of a dynamical system determine the evolution of the system. Typically an output is generated from the state of the system . This is a rather general definition of a dynamical system, where many different systems fit into. For investigating dynamical systems it is necessary to specify some characteristics that provide a subdivision into special classes of dynamical systems. Specific methods are available for some of these classes, thus such a classification can help to simplify the analysis.

An important characteristic of a dynamical system is whether it is continuous or discrete. Continuous systems (often called flows) are given by differential equations whereas discrete dynamical systems (often called maps) are specified by difference equations . Autonomous systems are characterized by the fact that input and output are omitted from the definition .

An important criterion for the analysis of a dynamical system is whether it is time-dependent or not . For time-dependent dynamical systems the function that specifies x (continuous case) orDelta x(discrete case) depends on the time itself whereas for time-independent systems this function does not change over time.

For the analysis it is very important whether a dynamical system is linear or not. Linear dynamical systems are simple to analyze as opposed to non-linear systems, which typically do have intricate dynamical behavior . Often linearization at specific locations is used to get insights into these complex non-linear dynamical systems.

Using linearization, another classification of dynamical systems is crucial to separate simple cases from more complex ones. Hyperbolic dynamical systems can be analyzed by linearization efficiently, whereas non-hyperbolic systems may cause major troubles in combination with linearization Hyperbolic systems are structurally stable, i.e., small perturbations of the system parameters do not change the qualitative behavior of the system. Non-hyperbolic systems are difficult to investigate, occur rarely and can be considered the transitional phase between two hyperbolic systems of different nature .

c.

The simplest population growth model, the Malthusian model, states that the rate of change of population is proportional to the population. In mathspeak we have

P'(t)=aP(t)

P(0)=P0

Here P(t) is the population at time t and a is a constant. (We assume that the population is a continuous function for simplicity. If we assume P(t) represents number of people, then obviously P(t) can take only integer values. A different type of analysis is required.)

This is an example of a linear separable ode. The exact solution to the problem is

P(t)=P0exp(at)

If a is positive, the populations grows exponentially for all time. This is unrealistic. A more realistic model is the logistic model

P'(t)=aP(t)(1-P(t)/b)

P(0)=P0

Here a and b are constants. In this model P'(t) is a sum of positive and negative terms (assuming P(t) is non-negative). If P(t) is sufficiently large, P'(t) is negative. It turns out as t increases the poplulation approaches b. b is the carrying capacity of this model.

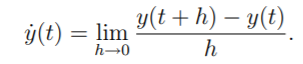
d.

The construction of numerical methods for initial value problems as well as basic properties of such methods shall first be explained for the simplest method: The explicit Euler method. Be aware that this method is not the most efficient one from the computational point of view. In later sections, when a basic understanding has been achieved, computationally efficient methods will be presented.

A general principle to derive numerical methods is to “discretize” constuctions like derivatives, integrals, etc. Given an initial value problem



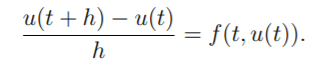
the operation which can not be evaluated numerically obviously is the limit h → 0, that defines the derivative



However, for any positive (small) h, the finite difference



can easily be evaluated. By definition, it is an approximation of the derivative y˙(t). Let us therefore approximate the differential equation ˙y(t) = f(t, y(t)) by the difference equation



Given u at time t, one can compute u at the later time t + h, by solving the difference equation



This is exactly one step of the explicit Euler method Introducing the notation tn+1 = tn + h and un= u(tn) it reads



The Runge-Kutta method attempts to overcome the problem of the Euler's method, as far as the choice of a sufficiently small step size is concerned, to reach a reasonable accuracy in the problem resolution.

On the other hand, the minus of the Runge-Kutta method is that we need to calculate the increments *ki* involved in the numerical algorithm, enlarging the utilized space of the Excel worksheet, generating a loss in efficiency, especially when, like in the Optimal Control theory, more variables are involved (i.e., the *state variable*, the *costate variable* and *the control variable*). To have things under a better control, when working with Excel, the goal is always to keep at minimum the worksheet space and the variables utilized, trying to maximize the results we want to achieve.

For the differential equation ẏ=f(t,y) where *y*(*t*0) = *y*0 the Runge-Kutta of fourth-order method (RK4) method is defined using the following recursion formula:

yn+1= yn +1/6(k1+2k2 +2k3 +2k4)

where:

1/6(k1+2k2 +2k3 +2k4) =weight average slope

K1 = f(tn,yn)

k2 = f(tn+h/2,yn+(h/2) k1)

k3 = f(tn+h/2,yn+(h/2) k2)

k4 = hf(tn+h,yn+hk3)

H= step size

Runge-Kutta methods of any order can be derived, although the derivation of an order higher than four can become extremely complicated.