PH5720

In this unit you will learn the determination of the numerical solution of differential equations using different approaches. To reduce the computation cost and achieve accurate numerical solution with small error, it is always advisable to compute the differential equation with larger stability region. You will study to solve the first order differential equations. There are several examples in the slide. You can write small program and see the numerical solution of the differential equation. The unit will focus on the following subsection.

- Ordinary Differential Equations: Initial Value Problems
- Ordinary Differential Equations: Boundary value Problems
- Partial Differential Equations

Book : Computational Physics by Morten Hjorth-Jensen

Reference material file name : lect_padhan_1_refrence material

Differential equations (DEs)

A differential equation (DE) is an equation involving a function and its derivatives.

$$\frac{dy}{dx} = f(x)$$

Where $f(x)$ is given and $y(x)$ is an unknown function of x.

When $f(x)$ is continuous over some interval, we found the general solution $y(x)$ by integration.

An **ordinary differential equation (ODE)** is a differential equation for a function of a single variable, e.g.,

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

A **partial differential equation (PDE)** is a differential equation for a function of several variables, e.g.,

$$\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial v}{\partial z} + \frac{\partial v}{\partial t} = v(x, y, z, t)$$

Differential equations (DEs)

A **differential equation (DE)** is an equation involving a function and its derivatives.

An **ordinary differential equation (ODE)** is a differential equation for a function of a single variable.

A **partial differential equation (PDE)** is a differential equation for a function of several variables.

The DE equation is linear if the source function f is linear on its second argument,

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

Linear DE

$$y' = f\big(t, y(t)\big) = 2y + 3$$

Non-linear DE

$$y' = f\big(t, y(t)\big) = -\frac{2}{ty} + 4t$$

Differential equations (DEs)

A **differential equation (DE)** is an equation involving a function and its derivatives.

An **ordinary differential equation (ODE)** is a differential equation for a function of a single variable.

A **partial differential equation (PDE)** is a differential equation for a function of several variables.

Linear DE **Non-linear DE**

A **homogeneous differential equation** has homogenous source function f of degree zero.

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

Inhomogeneous differential equation

$$y'' - 3y' - 4y = f(t)$$

homogeneous differential equation

$$y'' - 3y' - 4y = 0$$

Differential equations (DEs)

The order of a differential equation is the highest order derivative occurring in the equation.

$$x^{(k)}(t) = f(t, x', x'', \cdots\cdots\cdots x^{(k-1)})$$

k^{th} order DE

Define k new functions

$$u_1(t) = x(t), \qquad u_2(t) = x'(t), \qquad \cdots\cdots, u_k(t) = x^{(k-1)}(t)$$

Then original ODE is equivalent to **first-order system**

$$\begin{bmatrix} u_1'(t) \\ u_2'(t) \\ \vdots \\ u_{k-1}'(t) \\ u_k'(t) \end{bmatrix} = \begin{bmatrix} u_2(t) \\ u_3(t) \\ \vdots \\ u_k(t) \\ f(t, x', x'', \cdots\cdots\cdots x^{(k-1)}) \end{bmatrix}$$

Ordinary Differential equations (ODEs)

Order of ODE is determined by highest-order derivative of solution function appearing in ODE.

ODE with **higher-order derivatives** can be transformed into equivalent **first-order system**.

We will discuss numerical solution methods **only for first-order ODEs.**

Most ODE software is designed to solve only **first-order equations.**

Ordinary Differential Equations:

Differential equations (DEs)

Ordinary differential equation (ODE): all derivatives are with respect to single independent variable.

$$f(y)y' = g(x)$$

Solution of differential equation is a function in the infinite-dimensional space of functions.

$$\int f(y)y'\,dx = \int f(y)\frac{dy}{dx}dx = \int f(y)\,dy = \int g(x)\;dx$$

Numerical solution of the differential equations is based on the **finite-dimensional approximation**.

Differential equation is replaced by algebraic equation whose solution approximates that of given differential equation.

Ordinary differential equations (ODEs)

All derivatives are with respect to single independent variable, i.e., contain functions of one independent variable and derivatives in that variable.

Here are some examples…….

Newton's second law

Simple harmonic oscillations

Harmonic oscillations and external forces

Simple RLC circuit

The pendulum

ODEs Example

Contain functions of one independent variable and derivatives in that variable.

Newton's second law of motion : $F = ma$

Second order ODE

$$\frac{d^2x(t)}{dt^2} = \frac{F(x,t)}{m}$$

F and m are force and mass respectively.
$x(t)$ dependent variable and t independent variable

Defining $u_1 = x$ and $u_2 = x'$ yields equivalent system of two first order ODEs

$u_1 = x$: solution x of the original 2nd ODE
$u_2 = x'$: velocity

$$\begin{bmatrix} u_1' \\ u_2' \end{bmatrix} = \begin{bmatrix} u_1 \\ \frac{F(x,t)}{m} \end{bmatrix}$$

Can solve this by methods for 1st order ODE.

Ordinary differential equations (ODEs)

General first-order system of ODEs has the form

$$x'(t) = f(t, x)$$

Where, $x : \mathbb{R} \longrightarrow \mathbb{R}^n$

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}^{n+1}$$

$$x' = \frac{dx(t)}{dt}$$

$$\begin{bmatrix} x_1'(t) \\ x_2'(t) \\ \vdots \\ x_n'(t) \end{bmatrix} = \begin{bmatrix} \frac{dx_1(t)}{dt} \\ \frac{dx_2(t)}{dt} \\ \vdots \\ \frac{dx_n(t)}{dt} \end{bmatrix}$$

f : given, determine unknown x satisfying ODE

For special case of **single scalar ODE**, $n = 1$

ODEs Initial value problems

By itself, ODE $x'(t) = f(t, x)$ does **not determine unique solution function**.

ODE merely **specifies slope** $x'(t)$ of the solution function at each point, but **not actual value** at any point.

Infinite family of functions satisfies ODE, in general, provided f is sufficiently smooth.

To **single out particular solution**, value x_0 of solution function must be specified at some point t_0

Ordinary Differential Equations: Initial Value Problems

ODEs Initial value problems

By itself, ODE $x'(t) = f(t, x)$ does **not determine unique solution function**.

Thus, part of given problem data is requirement that

$$x(t_0) = x_0$$

which determine the unique solution to ODE.

Because of interpretation of independent variable t as the time, think as
t_0 : **initial time**
x_0 : **initial value**

Hence, this is termed as **initial value problem (IVP)**

ODE governs evolution of system in time from its initial state x_0 at time t_0 onward, and we **determine function $x(t)$** that describes state of system as a function of time.

ODEs Initial value problems : Example

Consider scalar ODE $x' = x$

Integration give : $x(t) = ce^t$, where integration constant c is any real constant

Imposing initial condition $x(t_0) = x_0$ singles out unique particular solution

Let, $t_0 = 0$, then $c = x_0$,

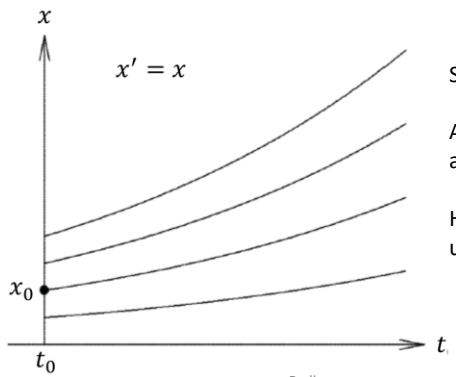
which means that the **solution** is : $\boldsymbol{x(t) = x_0 e^t}$

ODEs Initial value problems : Example

Consider scalar ODE $x' = x$

which means that the solution is : $\boldsymbol{x(t) = x_0 e^t}$

Family of solutions for ODE $x' = x$



Solution grow exponentially

Any two solution diverge away from each other

Hence, every solution is unstable

ODEs Initial value problems : Stability of solutions

Solution of ODE is
stable if solutions resulting from perturbations of initial value remain close to original solution.

asymptotically stable if solutions resulting from perturbations converge back to original solution.

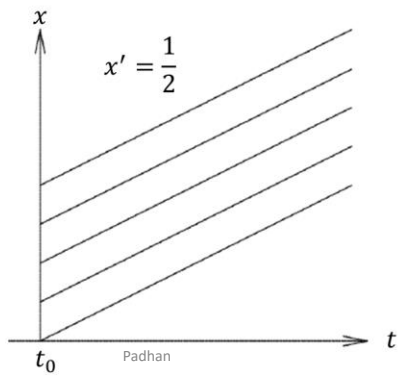
unstable if solutions resulting from perturbations diverge away from original solution without bound.

ODEs Initial value problems : Example, stable solutions

Consider scalar ODE $x' = \frac{1}{2}$ Let, $t_0 = 0$, then $c = x_0$,

which means that the solution is : $\boldsymbol{x(t) = \frac{t}{2} + x_0}$

Family of solutions for ODE $x' = \frac{1}{2}$

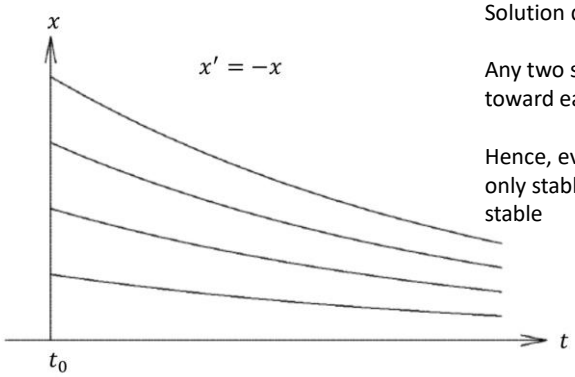


ODEs Initial value problems : Example, asymptotically stable solutions

Consider scalar ODE $x' = -x$ Let, $t_0 = 0$, then $c = x_0$,

which means that the solution is :$\boldsymbol{x(t) = x_0 e^{-t}}$

Family of solutions for ODE $x' = -x$



Solution decay exponentially

Any two solution converge toward each other

Hence, every solution is not only stable but asymptotically stable

ODEs Initial value problems: Examples, Stability of solutions

Consider scalar ODE $x' = \lambda x$, where λ is a constant

Let, t_0 : initial time
x_0 : initial value
which means that the solution is : $\boldsymbol{x(t) = x_0 e^{\lambda t}}$

For real λ
$\lambda > 0$: all non-zero solutions ***grow exponentially*** so every solution is unstable.
$\lambda < 0$: all non-zero solutions ***decay exponentially*** so every solution is not only stable, but asymptotically stable.

For complex λ
$Re\ (\lambda) > 0$: solution is **unstable**
$Re\ (\lambda) < 0$: solution is **asymptotically stable**
$Re\ (\lambda) = 0$: solution is **stable but not asymptotically stable**

ODEs Initial value problems: Example: linear systems

Linear, homogenous system of ODEs with constant coefficients has the form

$x' = Ax$, where A is a $n \times n$ matrix, initial condition is $\boldsymbol{x(0) = x_0}$

Suppose A is diagonalizable, with eigenvalues λ_i
and corresponding eigenvectors v_i
$i = 1, 2, \cdots\cdots\cdots\cdots, n$

Express x_0 as a linear combination $x_0 = \sum_{i=1}^{n} \alpha_i v_i$

which means that the solution is : $\boldsymbol{x(t) = \sum_{i=1}^{n} \alpha_i v_i e^{\lambda_i t}}$

ODEs IVP: Example, linear systems, stability of solutions

Linear, homogenous system of ODEs with constant coefficients has the form

$x' = Ax$, where A is a $n \times n$ matrix, initial condition is $\boldsymbol{x(0) = x_0}$

solution is : $\boldsymbol{x(t) = \sum_{i=1}^{n} \alpha_i v_i e^{\lambda_i t}}$

Eigenvalues of A : **positive real parts**
$\Longrightarrow$ solution is exponentially growing solution components
: **negative real part**
$\Longrightarrow$ solution is exponentially decaying solution components
: **zero real part** (i.e., pure imaginary)
$\Longrightarrow$ solution is oscillatory solution components

Solution : Stable if $Re(\lambda_i) \leq 0$ for every eigenvalue
Asymptotically stable if $Re(\lambda_i) \leq 0$ for every eigenvalue
Unstable if $Re(\lambda_i) > 0$ for any eigenvalue

ODEs Initial value problems : Numerical solution

Analytical solution of ODE is **closed form** formula that can be evaluated at any point 't'.

Numerical solution of ODE is **table of approximate values** of solution function at discrete set of points.

Numerical solution is **generated by simulating** behaviour of system governed by ODE.

Starting at t_0 with given initial value x_0, we track trajectory dictated by ODE.

Evaluating $f(t_0, y_0)$ tells us **slope of trajectory at that point**.

We use this information to predict value x_i of solution at future time $t_1 = t_0 + h$ for some suitably chosen time increment h.

ODEs Initial value problems : Numerical solution

Analytical solution of ODE is **closed form** formula that can be evaluated at any point 't'.

Evaluating $f(t_0, y_0)$ tells us **slope of trajectory** at that point.

We use this information to predict value x_i of solution at future time $t_1 = t_0 + h$ for some suitably chosen time **increment *h***.

Approximate solution values are generated **step by step in increments** moving across interval in which solution is sought.

In stepping from one discrete point to next, we **incur some error**, which means that next approximate solution value lies on **different solution from one we stared on**.

Stability or instability of solutions determines, in part, whether such errors are **magnified or diminished with time**.

Numerical solution of ordinary differential equation with Initial value problems :

Computational method

ODEs : Euler's method

For general system of ODEs $x' = f(t, x)$, consider Taylor series

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2}x''(t) + \cdots\cdots\cdots\cdots$$
$$x(t+h) = x(t) + hf(t,x) + \frac{h^2}{2}x''(t) + \cdots\cdots\cdots\cdots$$

Euler's method results from dropping terms of second and higher order to obtain approximate solution value.

$$x_{k+1} = x_k + h_k f(t_k, x_k)$$

Euler's method advances solution by **extrapolating along straight line** whose slope is given by $f(t_k, x_k)$

Euler's method is **single-step** method because it depends on information at only one point in time to advance to next point.

Solution if ODEs Initial value problems : Example, Euler's method

Consider ODEs $x' = x$,

Apply Euler's method to ODE

with step size : h

Advance solution from time $t_0 = 0$ to time $t_1 = t_0 + h$

$t_0 = 0 \implies x = x_0$

$$x_0' = \frac{x_1 - x_0}{t_1 - t_0} \Longrightarrow \frac{x_1 - x_0}{h} = x_0' \Longrightarrow x_1 = x_0 + hx_0' = x_0 + hx_0$$

$$x_1 = x_0(1 + h)$$

Value for solution we obtained at t_1 is not exact, $x_1 \neq x(t_1)$

For example, if $t_0 = 0, x_0 = 1$ and $h = 0.5$

Start $x_1 = 1.5$

$\boldsymbol{x(t) = x_0 e^t}$ $x(0.5) = e^{0.5} = 1.649$

Thus, x_1 : lies on different solution from one we started on.

Solution if ODEs Initial value problems : Example, Euler's method

Consider ODEs $x' = x$,

Apply Euler's method to ODE

with step size : h

Advance solution from time $t_0 = 0$ to time $t_1 = t_0 + h$

$$x_1 = x_0(1 + h)$$

For example, if $t_0 = 0$, $t_1 = t_0 + h, x_0 = 1$ and $h = 0.5$

Start $x_1 = 1.5$

$\boldsymbol{x(t) = x_0 e^t}$ $x(0.5) = e^{0.5} = 1.649$

Thus, x_1 : lies on different solution from one we started on.

For example, if $t_0 = 0$, $t_2 = t_1 + h, x_0 = 1$ and $h = 0.5$

Start $x_2 = 1.5(1 + 0.5) = 2.25$

Start $x_2 = 1.649(1 + 0.5) = 2.473$

$\boldsymbol{x(t) = x_0 e^t}$ $x(1) = e^1 = 2.718$

Thus, x_2 : lies on different solution from one we started on.

Continue . . . .

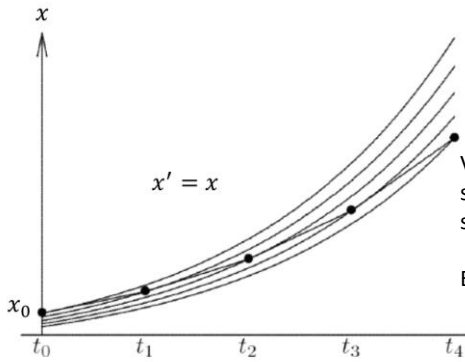
Solution if ODEs Initial value problems : Example, Euler's method

Consider ODEs $x' = x$,

Apply Euler's method to ODE, with step size : h

Advance solution from time $t_0 = 0$ to time $t_1 = t_0 + h$

$$x_1 = x_0(1 + h)$$



Value of x_1 lies on a different solution of the ODE from the starting one x_0

Error amplify with time

For unstable solutions, errors in numerical solution grow with time

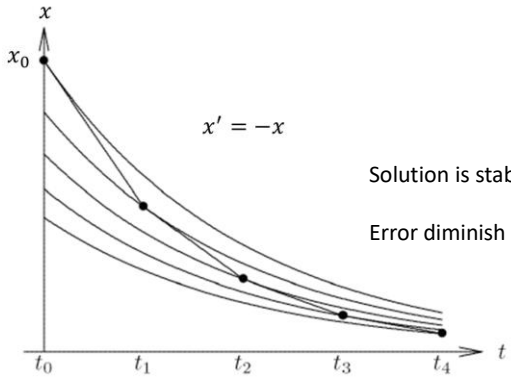
Solution if ODEs Initial value problems : Example, Euler's method

Consider ODEs $x' = -x$,

Apply Euler's method to ODE, with step size : h

Advance solution from time $t_0 = 0$ to time $t_1 = t_0 + h$

$$x_1 = x_0(1 + h)$$



Solution is stable

Error diminish with time

For stable solutions, errors in numerical solution may diminish with time

Numerical solution of ordinary differential equation with Initial value problems :

Computational method

Error in the numerical solution

Any errors introduced during the computation can be either amplified or diminished over time, depending on the stability of the solution sought.

Numerical errors in ODE solution

Numerical methods for solving ODEs incur two distinct types of error.

Rounding error, which is due to finite precision of floating-point arithmetic.

Truncation error (discretization error), which is due to approximation method used and would remain even if all arithmetic were exact.

In practice, truncation error is dominant factor determining accuracy of numerical solution of ODEs, so we will hence forth ignore rounding error.

ODEs Initial value problems [Numerical solution] : Global and local error

Truncation error at any point t_k can be expressed as the sum of
Global error and **Local error**

Global error: difference between computed solution and true solution $x(t)$ passing through the initial point (t_0, x_0).

$$e_k = x_k - x(t_k)$$

Local error: error made in one step of numerical method

$$\ell_k = x_k - u_{k-1}(t_k),$$

where $u_{k-1}(t_k)$ is true solution passing through previous point (t_{k-1}, x_{k-1})

ODEs Initial value problems [Numerical solution] : Global and local error

Global error: is not necessarily **sum of local error**.

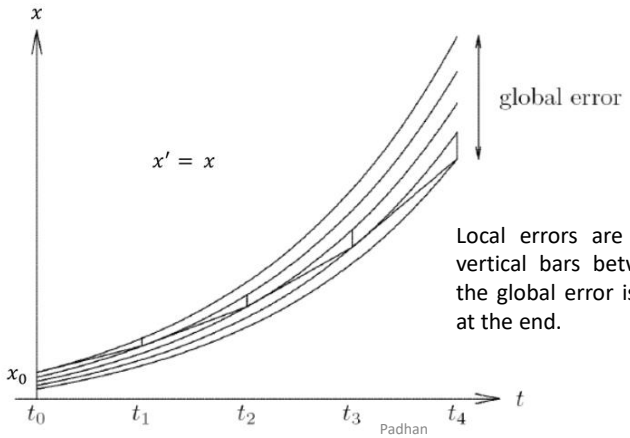
Global error: is generally greater than sum of local errors if solution are unstable, but may be less than sum if solution are stable.

Target is to have small global error, but we can control only local error directly.

ODEs Initial value problems [Numerical solution] : Global and local error

Global error: is not necessarily sum of local error.

Global error: is generally greater than sum of local errors if solution are unstable, but may be less than sum if solution are stable.

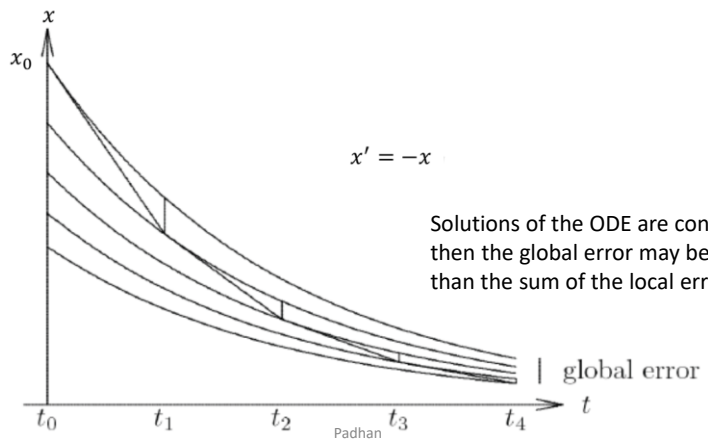


Local errors are indicated by small vertical bars between solutions and the global error is indicated by a bar at the end.

ODEs Initial value problems [Numerical solution] : Global and local error

Global error: is not necessarily sum of local error.

Global error: is generally greater than sum of local errors if solution are unstable, but may be less than sum if solution are stable.



Solutions of the ODE are converging, then the global error may be less than the sum of the local errors.

Numerical solution of ordinary differential equation with Initial value problems :

Computational method

Error in the numerical solution

Order of accuracy and stability
we need to characterize both its local error (accuracy)
and the compounding effects over multiple steps (stability)

ODEs IVP [Numerical solution] : Order of accuracy. Stability

Order of accuracy of numerical method is p if $\ell_k = \mathcal{O}(h_k^{p+1})$,

Then local error per unit step $\frac{\ell_k}{h_k} = \mathcal{O}(h_k^p)$

Under reasonable conditions, $e_k = \mathcal{O}(h_k^p)$, where h is average step size.

Numerical method is stable if small perturbations do not cause resulting numerical solutions to **diverge** from each other without bound.

Such divergence of numerical solutions could be **caused by instability** of solution to ODE, but can also be due to **numerical method itself**, even when solutions to ODE are stable.

ODEs IVP [Numerical solution] : Determining stability/accuracy

Simple approach to determining stability and accuracy of numerical method is to apply it to scalar ODE $x' = \lambda x$, where λ is (possibly complex) constant

Let, t_0 : initial time
x_0 : initial value
which means that the solution is : $\boldsymbol{x(t) = x_0 e^{\lambda t}}$

Determine stability of numerical method by characterizing growth of numerical solution.

Determine accuracy of numerical method by comparing exact and numerical solutions.

ODEs IVP [Numerical solution] : Determining stability/accuracy

Example: Euler's method

ODE $x' = \lambda x$, where λ is (possibly complex) constant

Let, t_0 : initial time
x_0 : initial value
which means that the solution is : $\boldsymbol{x(t) = x_0 e^{\lambda t}}$

$$x_{k+1} = x_k + h\lambda x_k = (1 + h\lambda)x_k$$

$x_1 = (1 + h\lambda)x_0, \quad x_2 = (1 + h\lambda)^2 x_0, \quad x_3 = (1 + h\lambda)^3 x_0, \cdots\cdots$

Which means that $x_k = (1 + h\lambda)^k x_0$

If $Re(\lambda) < 0$, exact solution decays to zero as t increases as does computed solution if $|1 + h\lambda| < 1$, which holds if $h\lambda$ lies outside circle in complex plane of radius centered at -1.

ODEs IVP [Numerical solution] : Determining stability/accuracy

Example: Euler's method

ODE $x' = \lambda x$, where λ is (possibly complex) constant

Solution is : $\boldsymbol{x(t) = x_0 e^{\lambda t}}$

$$x_{k+1} = x_k + h\lambda x_k = (1 + h\lambda)x_k \Longrightarrow \boldsymbol{x_k = (1 + h\lambda)^k x_0}$$

If $Re(\lambda) < 0$, exact solution decays to zero as t increases as does computed solution if $|1 + h\lambda| < 1$, which holds if $h\lambda$ lies outside circle in complex plane of radius centered at -1.

If λ is real, then $h\lambda$ must lie in interval $(-2, 0)$, so for $\lambda < 0$, we must have $h \leq -\frac{2}{\lambda}$ for Euler's method to be stable.

Growth factor $1 + h\lambda$ agrees with series expansion

$$\boldsymbol{e^{\lambda t} = 1 + h\lambda + \frac{(h\lambda)^2}{2} + \frac{(h\lambda)^3}{6} + \cdots\cdots}$$

through terms of first order in h, so Euler's method is first-order accurate

ODEs : Euler's method

For general system of ODEs $x' = f(t, x)$, consider Taylor series

$$x(t + h) = x(t) + hx'(t) + \mathcal{O}(h^2)$$
$$x(t + h) = x(t) + hf(t, x) + \mathcal{O}(h^2)$$

For $t = t_k$ and $h = h_k$

$$x(t_{k+1}) = x(t_k) + h_k f\big(t_k, x(t_k)\big) + \mathcal{O}\big(h_k^2\big)$$

Euler's method

$$x_{k+1} = x_k + h_k f(t_k, x_k)$$

Subtracting

$$e_{k+1} = x_{k+1} - x(t_{k+1})$$
$$= [x_k - x(t_k)] + h_k\big[f(t_k, x_k) - f\big(t_k, x(t_k)\big)\big] - \mathcal{O}\big(h_k^2\big)$$

ODEs : Euler's method

For general system of ODEs $x' = f(t, x)$, consider Taylor series
For $t = t_k$ and $h = h_k$

$$x(t_{k+1}) = x(t_k) + h_k f\big(t_k, x(t_k)\big) + \mathcal{O}\big(h_k^2\big)$$

Euler's method

$$x_{k+1} = x_k + h_k f(t_k, x_k)$$

Subtracting

$$e_{k+1} = x_{k+1} - x(t_{k+1})$$
$$= [x_k - x(t_k)] + h_k\big[f(t_k, x_k) - f\big(t_k, x(t_k)\big)\big] - \mathcal{O}\big(h_k^2\big)$$

If there were no prior errors, then we would have $x_k = x(t_k)$, and differences in bracket on the right side would be zero, leaving only $\mathcal{O}\big(h_k^2\big)$ term, which is local error.

This means that Euler's method is first-order accurate.

ODEs : Euler's method

For general system of ODEs $x' = f(t, x)$,

$$e_{k+1} = x_{k+1} - x(t_{k+1})$$
$$= [x_k - x(t_k)] + h_k\big[f(t_k, x_k) - f\big(t_k, x(t_k)\big)\big] - \mathcal{O}\big(h_k^2\big)$$

$\mathcal{O}\big(h_k^2\big)$ is local error.
Global error is sum of **local error** and **propagated error**

From **mean value theorem**

$$f(t_k, x_k) - f\big(t_k, x(t_k)\big) = \boldsymbol{J_f}(t_k, \boldsymbol{\xi})\big(x_k - x(t_k)\big)$$

For some value of ξ, where J_f is Jacobian matrix of f with respect to x.

So, global error at step $k+1$ is $\quad e_{k+1} = \big(\boldsymbol{I} + h_k J_f\big)e_k + \ell_{k+1}$

Thus, global error is multiplied at each step by **growth factor** $I + h_k J_f$

Error do not grow if spectral radius $\boldsymbol{\rho}\big(I + h_k J_f\big) \leq 1$, which holds if all eigenvalues of $h_k J_f$ lie inside circle in complex plane of radius 1 centered at -1

Stability in ODE, in general

In general, **growth factor depends** on

Numerical method, which determines form of growth factor

Step size h

Jacobian J_f, which is determined by particular ODE

Numerical solution of ordinary differential equation with Initial value problems :

Computational method

Error in the numerical solution

Order of accuracy and stability

Control of growth factor

Step size selection for the numerical solution of ODE

In choosing **step size** for advancing numerical solution of ODE, we want to take **large steps** to **reduce computational cost**, but must also take into account both **stability and accuracy**

To yield meaningful solution, step size must obey any **stability** restrictions

In addition, local error estimate is needed to ensure that desired **accuracy** is achieved

With Euler's method, for example, local error is approximately $\left(\frac{h_k^2}{2}\, x''\right)$, so choose step size to satisfy

$$h_k \leq \sqrt{\frac{2\ \ tolerance}{|x''|}}$$

Given a tolerance for the maximal local error.

Step size selection for the numerical solution of ODE

With Euler's method, for example, local error is approximately $\left(\frac{h_k^2}{2}\, x''\right)$, so choose step size to satisfy

$$h_k \leq \sqrt{\frac{2\ \ tolerance}{|x''|}}$$

We do not know value of x'', but can estimate it by difference quotient

$$x'' \approx \frac{x_k' - x_{k-1}'}{t_k - t_{k-1}}$$

Other methods of obtaining error estimates are based on difference between results obtained using methods of **different orders** or **different step sizes**.

Numerical solution of ordinary differential equation with Initial value problems :

Computational method

Error in the numerical solution

Order of accuracy and stability

Control of growth factor

Implicit methods

Implicit methods

Euler's method is **explicit**.
It uses only information at t_k to advance solution to time t_{k+1}

This may be desirable, but Euler's method has rather **limited stability region**

Larger stability region can be obtained by using information at time t_{k+1}, which makes the method **implicit**.

Simplest example is **backward Euler method**

$$x_{k+1} = x_k + h_k f(t_{k+1}, x_{k+1})$$

Method is **implicit because** we must evaluate f with argument x_{k+1} before we know its value.

Implicit methods

Euler's method is **explicit**.

Larger stability region can be obtained by using information at time t_{k+1}, which makes the method **implicit**.

Simplest example is **backward Euler method**

$$x_{k+1} = x_k + h_k f(t_{k+1}, x_{k+1})$$

Method is **implicit because** we must evaluate f with argument x_{k+1} before we know its value.

We must solve algebraic equation to determine x_{k+1}.

Typically we use iterative method such as Newton's method or fixed iteration to solve for x_{k+1}.

Good starting guess for iteration can be obtained from **explicit method**, such as Euler's method, or from solution at previous time step.

Implicit methods: Backward Euler method, Example

Consider nonlinear scalar ODE

$$x' = -x^3 \quad and \quad x(0) = 1$$

Use backward Euler's method with step size $h = 0.5$.

$$x_1 = x_0 + hf(t_1, x_1) = 1 - 0.5x_1^3$$

Nonlinear equation for x_1 could be solved by fixed-point iteration or Newton's method.

To obtained starting guess for x_1

Use previous solution value $x_0 = 1$

or use explicit method such as Euler's method : $x_1 = x_0 - 0.5\, x_0^3 = 0.5$

Iteration eventually converge to final value $x_1 \approx 0.7709$

Numerical solution of ordinary differential equation with Initial value problems :

Computational method

Error in the numerical solution

Order of accuracy and stability

Control of growth factor

Implicit methods

Advantage: Implicit methods generally have significantly larger stability region than comparable explicit methods

Implicit methods: Backward Euler method, Example

Consider scalar ODE $x' = \lambda x$, where λ is (possibly complex) constant

Let, t_0 : initial time and x_0 : initial value

which means that the solution is : $\boldsymbol{x(t) = x_0 e^{\lambda t}}$

$$x_{k+1} = x_k + h\lambda x_{k+1} \implies x_k = (1 - h\lambda)x_{k+1}$$

$$x_k = \left(\frac{1}{1 - h\lambda}\right)^k x_0$$

Thus, for backward Euler to be stable, we must have

$$\left|\frac{1}{1 - h\lambda}\right| \leq 1,$$

which hold for any $h > 0$ when $Re(\lambda) < 0$.

So stability region for backward Euler method includes entire left half of complex plane, or interval $(-\infty, 0)$ if λ is real.

Implicit methods: Backward Euler method, Example

Growth factor

$$\frac{1}{1-h\lambda} = 1 + h\lambda + (h\lambda)^2 + \cdots\cdots,$$

agrees with expansion for $e^{\lambda t}$ through terms of order h, so backward Euler method is first-order accurate.

Growth factor of backward Euler method for general system of ODEs $x' = f(t, x)$, is $\left(I + h_k J_f\right)^{-1}$, whose spectral radius is less than 1 provided all eigenvalues of hJ_f lie outside circle in complex plane of radius 1 centered at 1.

Thus, stability region of backward Euler for general system of ODEs is entire left half of complex plane.

Implicit methods: Backward Euler method, Example

Thus, for computing stable solution backward Euler is stable for any positive step size, which means that it is **unconditionally stable**.

Great virtue of unconditionally stable method is that desired accuracy is only constraint on choice of step size.

Thus, we may be able to take much larger steps than for explicit method of comparable order and attain much higher overall efficiency despite requiring more computation per step.

Although backward Euler method is unconditionally stable, its accuracy is only of first order, which severally limits its usefulness.