# MA5233 Computational Mathematics

## Lecture 10: Runge-Kutta Methods

Simon Etter

National University of Singapore

Semester I, AY 2020/2021

# Runge-Kutta Methods

**Introduction**

The aim of this lecture is to develop numerical methods for solving problems of the following form.

**Def: Ordinary differential equation (ODE)**

Given $f : \mathbb{R}^n \to \mathbb{R}^n$, $y_0 \in \mathbb{R}^n$ and $T > 0$, find $y : [0, T) \to \mathbb{R}^n$ such that

$$y(0) = y_0 \qquad \text{and} \qquad \dot{y}(t) = f\big(y(t)\big) \quad \text{for all } t \in [0, T).$$

$\dot{y} = y' = \frac{dy}{dt}$ is a shorthand notation for taking derivatives of a function $y : \mathbb{R} \to \mathbb{R}^n$.

**Outlook**

The following slides illustrate the above definition by discussing three example ODEs. These ODEs will reappear later in this lecture, so pay attention even if you are already familiar with ODEs.

# Runge-Kutta Methods

**Example 1**

Consider the problem of finding $y : [0, \infty) \to \mathbb{R}$ such that

$$y(0) = y_0 \qquad \text{and} \qquad \dot{y}(t) = \lambda\, y(t)$$

for some given $y_0, \lambda \in \mathbb{R}$.

The solution to this problem is given by

$$y(t) = y_0 \, \exp(\lambda t)$$

because this function satisfies

$$y(0) = y_0 \, \exp(\lambda 0) = y_0 \qquad \text{and} \qquad \dot{y}(t) = y_0 \, \exp(\lambda t)\, \lambda = \lambda\, y(t).$$

# Runge-Kutta Methods

**Example 2**

Consider the problem of finding $y : [0, \frac{1}{y_0}) \to \mathbb{R}$ such that

$$y(0) = y_0 \qquad \text{and} \qquad \dot{y}(t) = y(t)^2$$

for some given $y_0 \in \mathbb{R}$.

The solution to this problem is given by

$$y(t) = \frac{y_0}{1 - y_0 t}$$

because this function satisfies

$$y(0) = \frac{y_0}{1 - y_0\, 0} = y_0 \qquad \text{and} \qquad \dot{y}(t) = \frac{y_0^2}{(1 - y_0 t)^2} = y(t)^2.$$

Note that this $y(t)$ is undefined for $t = \frac{1}{y_0}$. This shows that unlike the solution of Example 1, the solution to this ODE has a finite domain of definition $[0, \frac{1}{y_0})$.

The formula for $y(t)$ is well defined for $t > \frac{1}{y_0}$, but this part of the solution has no mathematical meaning since it is disconnected from the starting point $t = 0$.

# Runge-Kutta Methods

**Example 3**

Consider the problem of finding $x : [0, \infty) \to \mathbb{R}$ such that

$$x(0) = 1 \qquad \text{and} \qquad \ddot{x} = -x.$$

The solution to this equation is given by

$$x(t) = \cos(t)$$

since this function satisfies $x(0) = \cos(0) = 1$ and

$$\ddot{x}(t) = \tfrac{d^2}{dt^2}\cos(t) = -\tfrac{d}{dt}\sin(t) = -\cos(t) = -x(t).$$

$\ddot{x} = -x$ is not of the form $\dot{y} = f(y)$, but it can easily be reduced to this form: if we set

$$y = \left(\begin{smallmatrix} x \\ \dot{x} \end{smallmatrix}\right) \qquad \text{and} \qquad f(y) = \left(\begin{smallmatrix} y[2] \\ -y[1] \end{smallmatrix}\right),$$

then this $y(t)$ and $f(y)$ indeed satisfy

$$\dot{y} = \left(\begin{smallmatrix} \dot{x} \\ \ddot{x} \end{smallmatrix}\right) = \left(\begin{smallmatrix} y[2] \\ -y[1] \end{smallmatrix}\right) = f(y).$$

This shows that considering ODEs involving only first-order derivatives is enough to cover ODEs involving arbitrarily high derivatives.

# Runge-Kutta Methods

**Real-world example: Newton's law of motion**

Consider a point particle of mass $m > 0$ subject to a force $F(x) \in \mathbb{R}^3$ which depends on the location $x \in \mathbb{R}^3$ of the particle.

Newton's law of motion then states that the trajectory $x(t) \in \mathbb{R}^3$ of this particle satisfies

$$m\,\ddot{x}(t) = F\big(x(t)\big).$$

This equation is again not of the form $\dot{y} = f(y)$ but can reduced to this form by setting

$$y = \begin{pmatrix} x \\ \dot{x} \end{pmatrix}, \qquad f(y) = \begin{pmatrix} y[2] \\ \frac{1}{m}\,F(y[1]) \end{pmatrix}$$

such that

$$\dot{y} = \begin{pmatrix} \dot{x} \\ \ddot{x} \end{pmatrix} = \begin{pmatrix} \dot{x} \\ \frac{1}{m}\,F(x) \end{pmatrix} = \begin{pmatrix} y[2] \\ \frac{1}{m}\,F(y[1]) \end{pmatrix} = f(y).$$

# Runge-Kutta Methods

**Time-dependent ODEs**

Many textbooks allow the ODE-defining function $f(y, t)$ to also depend on the time variable $t$. I omit this generality here because it is rarely needed in applications and it significantly complicates the analysis.

# Runge-Kutta Methods

**ODEs vs PDEs**

ODEs are similar to PDEs in that the problem is to find a function $y(t)$ given an equation in terms of $y(t)$ and its derivatives which is to hold at every point in the domain.

Historically / formally, the defining property of an ODE is that the unknown $y(t)$ depends on only a single variable and hence the derivatives in the differential equations are *ordinary* derivatives.

By contrast, the unknown $u(x_1, \dots, x_d)$ in a PDE depends on several variables and hence the derivatives are *partial* derivatives.

The terms "ODE" and "PDE" are hardly ever used in this way anymore, however.

In modern terminology, the defining property of an ODE is that fixed values for $y(t)$ and its derivatives are specified at a single point $t_0$. For this reason, ODEs are also called *initial value problems*.

The defining property of a PDE is that fixed values of $u(x)$ and its derivatives are specified at two or more points $x \in \partial\Omega$. For this reason, PDEs are also called *boundary value problems*.

# Runge-Kutta Methods

**Example: ODEs vs PDEs**

The one-dimensional Poisson equation $-u''(x) = f(x)$ is an ODE in the formal sense because there is only a single independent variable $x$.

However, this equation is usually called a PDE because it is almost always paired with boundary conditions rather than initial conditions and hence it is much more similar to e.g. the higher-dimensional Poisson equation $-\Delta u = f$ than to Newton's law of motion $m\,\ddot{x} = F(x)$.

# Runge-Kutta Methods

**Discussion**

As mentioned earlier, our goal in this lecture is to develop numerical algorithms which approximate the map

$$\big(f(y), y_0, T\big) \quad \mapsto \quad y(t) \quad \text{such that} \quad y(0) = y_0, \quad \dot{y}(t) = f(y(t)).$$

However, it is advisable to first establish that this map is well defined and well conditioned, since otherwise we might end up constructing (and trusting!) numerical solutions $\tilde{y}(t)$ which have no meaning.

It turns out that the key condition guaranteeing the existence of solutions $y(t)$ is Lipschitz continuity as defined on the next slide.

# Runge-Kutta Methods

**Def: (Global) Lipschitz continuity**

A function $f : D \to \mathbb{R}^n$ with $D \subset \mathbb{R}^n$ is called *(globally) Lipschitz continuous with Lipschitz constant* $L > 0$ if for all $y_1, y_2 \in D$ we have

$$\left\| f(y_1) - f(y_2) \right\| \leq L \left\| y_1 - y_2 \right\|.$$

Note that whether a function $f(y)$ is Lipschitz continuous does not depend on the choice of norm due to norm equivalence in finite dimensions. However, the Lipschitz constant $L$ may depend on the choice of norm.

**Def: Local Lipschitz continuity**

A function $f : \mathbb{R}^n \to \mathbb{R}^n$ is called *locally Lipschitz continuous* if for every $y_0 \in \mathbb{R}^n$ there exists a pair $\delta, L > 0$ such that for all $y_1, y_2 \in \mathbb{R}^n$ we have

$$\left\| y_k - y_0 \right\| \leq \delta \qquad \Longrightarrow \qquad \left\| f(y_1) - f(y_2) \right\| \leq L \left\| y_1 - y_2 \right\|.$$

# Runge-Kutta Methods

**Discussion**

Lipschitz continuity of a function $f(y)$ is usually most conveniently verified using the following result.

**Corollary**

Assume $D \subset \mathbb{R}^n$ is convex and $f : D \to \mathbb{R}^n$ is differentiable everywhere in $D$. Then, $f(y)$ is locally Lipschitz continuous, and it is globally Lipschitz continuous if $\|\nabla f\|$ is bounded.

*Proof.* Immediate corollary of the result on the next slide.

# Runge-Kutta Methods

**Lemma: Lipschitz constants and derivatives**

Assume $D \subset \mathbb{R}^n$ is convex and $f : D \to \mathbb{R}^n$ has a bounded derivative. Then,

$$\|f(y_1) - f(y_2)\| \leq L\,\|y_1 - y_2\| \qquad \text{where} \qquad L = \sup_{y \in D} \|\nabla f(y)\|$$

*Proof.* According to the chain rule, we have that

$$\frac{d}{dt}\Big(f\left(y_1 + t\left(y_2 - y_1\right)\right)\Big) = \nabla f\left(y_1 + t\left(y_2 - y_1\right)\right)\left(y_2 - y_1\right)$$

and hence we conclude using the fundamental theorem of calculus that

$$\begin{aligned}
\|f(y_1) - f(y_2)\| &= \left\| \int_0^1 \nabla f\left(y_1 + t\left(y_2 - y_1\right)\right)\left(y_2 - y_1\right) dt \right\| \\
&\leq \int_0^1 \left\|\nabla f\left(y_1 + t\left(y_2 - y_1\right)\right)\right\| \left\|y_2 - y_1\right\| dt \\
&\leq \left(\sup_{y \in D} \|\nabla f(y)\|\right) \left\|y_2 - y_1\right\|.
\end{aligned}$$

# Runge-Kutta Methods

**Discussion**

Armed with the definition of Lipschitz continuity, we can now establish that the ODE function $\big(f(y), y_0, T\big) \mapsto y(t)$ is indeed well defined if $f(y)$ is Lipschitz continuous.

**Picard-Lindelöf theorem**

Assume $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous. Then, there exists a unique function $y : [0, \infty) \to \mathbb{R}^n$ such that

$$y(0) = y_0 \qquad \text{and} \qquad \dot{y}(t) = f\big(y(t)\big) \quad \text{for all } t \in [0, \infty).$$

*Proof.* Beyond the scope of this module.

**Example**

Consider the function $f(y) = \lambda y$ from the example on slide 3. Since $f'(y) = \lambda$ is bounded for all $y$, this function is globally Lipschitz continuous and hence the solution $y(t) = y_0 \exp(\lambda t)$ exists for all $t \geq 0$.

# Runge-Kutta Methods

**Discussion**

The above result assumes global Lipschitz continuity of $f(y)$ but in returns guarantees that the solution is well defined for all times $t \geq 0$.

If $f(y)$ is only local Lipschitz, then the following result establishes that the solution $y(t)$ is well defined at least on some interval $[0, T] \subset [0, \infty)$.

**Picard-Lindelöf theorem, local version**

Assume $f : \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitz continuous. Then, there exists a $T > 0$ and a unique function $y : [0, T) \to \mathbb{R}^n$ such that

$$y(0) = y_0 \qquad \text{and} \qquad \dot{y}(t) = f\big(y(t)\big) \quad \text{for all } t \in [0, T).$$

*Proof.* Beyond the scope of this module.

**Example**

Consider the function $f(y) = y^2$ from the example on slide 4. Since $f'(y) = 2y$ exists but is unbounded $\mathbb{R}$, this function is locally Lipschitz but not globally Lipschitz. Correspondingly, the solution $y(t) = \frac{y_0}{1 - y_0 t}$ is defined only on the finite interval $[0, \frac{1}{y_0})$ but not on all of $[0, \infty)$.

# Runge-Kutta Methods

**Discussion**

In order to approximate the ODE map $\big(f(y), y_0, T\big) \mapsto y(t)$ numerically, we need this map to be not only well defined but also continuous with respect to $y_0$; if this is not the case, then any small perturbation in $y_0$ (e.g. floating-point rounding) may lead to arbitrarily large errors in the solution $y(t)$.

We will see on the next slide that continuity is a simple consequence of the following auxiliary result.

**Gronwall's inequality**

$$\dot y(t) \leq \lambda\, y(t) \qquad \Longrightarrow \qquad y(t) \leq \exp(\lambda t)\, y(0).$$

*Proof.* Consider $z(t) = \exp(-\lambda t)\, y(t)$. Then, $z(0) = y(0)$ and

$$\begin{aligned} \dot z(t) &= -\lambda \exp(-\lambda t)\, y(t) + \exp(-\lambda t)\, \dot y(t) \\ &\leq -\lambda \exp(-\lambda t)\, y(t) + \exp(-\lambda t)\, \lambda\, y(t) = 0; \end{aligned}$$

hence $z(t) \leq y(0)$ and thus $y(t) \leq y(0) \exp(\lambda t)$.

# Runge-Kutta Methods

**Thm: Lipschitz continuity of the ODE map**

Assume $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous with Lipschitz constant $L$, and assume $y_1, y_2 : [0, T) \to \mathbb{R}^n$ satisfy $\dot{y}_k = f(y_k)$. Then,

$$\|y_1(t) - y_2(t)\| \le \exp(Lt)\, \|y_1(0) - y_2(0)\| \qquad \text{for any } t < T.$$

*Proof.* We have for any $y : \mathbb{R} \to \mathbb{R}^n$ that

$$\tfrac{d}{dt}\|y(t)\| = \lim_{\tilde{t} \to t} \tfrac{\|y(\tilde{t})\| - \|y(t)\|}{\tilde{t} - t} \le \lim_{\tilde{t} \to t} \tfrac{\|y(\tilde{t}) - y(t)\|}{\tilde{t} - t} = \|\dot{y}(t)\|.$$

Combining the above with the Lipschitz continuity of $f(y)$, we obtain

$$\begin{aligned} \tfrac{d}{dt}\|y_2(t) - y_1(t)\| &\le \|\dot{y}_2(t) - \dot{y}_1(t)\| \\ &= \|f(y_2(t)) - f(y_1(t))\| \\ &\le L\, \|y_2(t) - y_1(t)\| \end{aligned}$$

from which the claim follows by Gronwall's inequality.

# Runge-Kutta Methods

**Discussion**

The bound on the previous slide is a two-sided coin:

- Good: error at time $t$ is proportional to error at time 0.
- Bad: constant of proportionality is $\exp(Lt)$.

The exponential function has a noticeable two-stage character:

- For $t \lesssim \frac{1}{L}$, $\exp(Lt)$ is about 1.
- For $t \gtrsim \frac{1}{L}$, $\exp(Lt)$ grows very quickly.

In practice, this means that there is often a characteristic time-scale $\frac{1}{L}$ beyond which numerical simulations become unreliable.

For example, we can predict the weather fairly accurately for the next 2-3 days, but predictions beyond one week are virtually impossible.

# Runge-Kutta Methods

**Solving ODEs using quadrature**

We have now established that if $f(y)$ is Lipschitz continuous, then $\dot{y} = f(y)$ has a unique solution and this solution is a Lipschitz continuous function of the initial conditions $y_0$.

Let us now move on to discuss numerical methods for approximating $y(t)$.

According to the fundamental theorem of calculus, we have

$$\left.\begin{aligned} y(0) &= y_0 \\ \dot{y} &= f(y) \end{aligned}\right\} \qquad \iff \qquad y(t) = y_0 + \int_0^t f\big(y(\tau)\big)\, d\tau.$$

Given a quadrature formula $(\theta_k, w_k)_{k=1}^s$ for $[0,1]$, we can hence compute a numerical approximation to $y(t)$ using the formula

$$\tilde{y}(t) = y_0 + \sum_{k=1}^{s} f\big(y(\theta_k t)\big)\, w_k t.$$

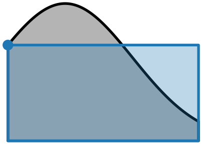
The quadrature points $\theta_k$ and weights $w_k$ are multiplied by $t$ because we need to scale the quadrature rule from $[0,1]$ to $[0,t]$.

However, we need to be careful because we do not know $y(t)$ for $t > 0$ and hence we cannot evaluate $f\big(y(\theta_k t)\big)$ for quadrature points $\theta_k > 0$.

# Runge-Kutta Methods

**Solving ODEs using quadrature (continued)**

The above suggests that we solve ODEs using the 1-point quadrature rule

$$\theta_1 = 0, \qquad w_1 = 1 \qquad \longleftrightarrow$$  .

This yields the following ODE solver.

**Def: Euler step**

Approximate the solution to $\dot{y} = f(y)$ using

$$y(t) \approx y(0) + f\big(y(0)\big)\, t.$$

**Discussion**

The error of this formula is easily estimated using Taylor's theorem.

# Runge-Kutta Methods

**Thm: Error estimate for Euler step**

Assume $f(y)$ is differentiable, $y(t)$ satisfies $\dot{y} = f(y)$, and $t > 0$. Then, there exists $\xi \in (0, t)$ such that

$$y(t) \;=\; \underbrace{y(0) + f(y(0))\, t}_{\text{Euler step}} \;+\; \underbrace{\tfrac{1}{2}\, \nabla f(\xi)\, \dot{y}(\xi)\, t^2}_{\text{error}}\,.$$

*Proof.* According to Taylor's theorem, there exists $\xi \in (0, t)$ such that

$$y(t) = y(0) + \dot{y}(0)\, t + \tfrac{1}{2}\, \ddot{y}(\xi)\, t^2.$$

The claim follows by noting that since $\dot{y}(t) = f(y(t))$, we have

$$\dot{y}(0) = f(y(0)) \qquad \text{and} \qquad \ddot{y}(\xi) = \nabla f(y(\xi))\, \dot{y}(\xi).$$

# Runge-Kutta Methods

**Discussion**

The error estimate

$$y(t) \;=\; \underbrace{y(0) + f\big(y(0)\big)\,t}_{\text{Euler step}} \;+\; \underbrace{\tfrac{1}{2}\,\nabla f(\xi)\,\dot{y}(\xi)\,t^2}_{\text{error}}$$

gives us two important pieces of information.

- The error is smaller the closer $f(y)$ and $y(t)$ are to being constant (i.e. the smaller their derivative).

  This is not surprising since the quadrature rule $\theta_1 = 0$, $w_1 = 1$ is exact for constant integrands.

- The error is smaller for smaller times $t$.

  This is again not surprising because for smaller $t$, the integral $\int_0^t f\big(y(\tau)\big)\,d\tau$ is both smaller and the integrand $f\big(y(\tau)\big)$ is closer to being constant over the interval $[0, t]$.

Both the ODE function $f(y)$ and the time interval $[0, T]$ are imposed on us by the application; hence the Euler step presented above gives us no means to improve the accuracy by investing more compute time.

This circumstance can be remedied using the composition idea presented on the next slide.

# Runge-Kutta Methods

**Def: Temporal mesh**

An ordered sequence of points $0 = t_0 < t_1 < \ldots$.

**Def: Euler's method**

Approximate the solution to $\dot y = f(y)$ using

$$\tilde y(0) = y(0), \qquad \tilde y\big(t\big) = \tilde y(t_{k-1}) + f\big(\tilde y(t_{k-1})\big)\, t \quad \text{for } t \in (t_{k-1}, t_k],$$

where $(t_k)_k$ denotes a given temporal mesh.

**Numerical demonstration**

See `euler_step()`, `integrate()` and `example()`.

**Discussion**

We intuitively expect that composing Euler steps into Euler's method yields a numerical approximation $\tilde y(t)$ which converges to the exact solution $y(t)$ as the mesh width $\max_k |t_k - t_{k-1}|$ goes to 0.

I will next show that this is indeed the case, and I will do so using the notions of exact and Euler time propagators introduced on the next slide.

# Runge-Kutta Methods

**Def: (Exact) time propagator**

The *(exact) time propagator* associated with $f : \mathbb{R}^n \to \mathbb{R}^n$ is the function

$$\Phi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n, \qquad \Phi(y_0, t) = y(t)$$

where $y(t)$ is the solution to $\dot{y} = f(y)$, $y(0) = y_0$.

**Def: Euler time propagator**

The *Euler time propagator* associated with $f : \mathbb{R}^n \to \mathbb{R}^n$ is the function

$$\tilde{\Phi} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n, \qquad \tilde{\Phi}(y_0, t) = y_0 + f(y_0)\, t.$$

**Shorthand notations**

In the following, I will use the shorthand notations

$$y_k = y(t_k), \qquad \tilde{y}_k = \tilde{y}(t_k),$$

and

$$\Phi_k(y_0) = \Phi\big(y_0, t_k - t_{k-1}\big), \qquad \tilde{\Phi}_k(y_0) = \tilde{\Phi}\big(y_0, t_k - t_{k-1}\big),$$

where $(t_k)_k$ denotes some temporal mesh specified by the context.

# Runge-Kutta Methods

**Thm: Error estimate for Euler's method**

Denote by $y(t)$ the solution to $\dot{y} = f(y)$ and by $\tilde{y}(t)$ the numerical approximation obtained using Euler's method with mesh $(t_k)_{k=0}^{n}$. Then,

$$\left\|\tilde{y}_n - y_n\right\| \;\leq\; \sum_{k=1}^{n} \exp\big(L\,(t_n - t_k)\big)\;\left\|\tilde{\Phi}_k(\tilde{y}_{k-1}) - \Phi_k(\tilde{y}_{k-1})\right\|,$$

assuming $f(y)$ is Lipschitz continuous with Lipschitz constant $L$.

*Proof.*

$$\begin{aligned}
\|\tilde{y}_n - y_n\| &= \|\tilde{\Phi}_n(\tilde{y}_{n-1}) - \Phi_n(y_{n-1})\| && \text{(definition of time propagators)}\\
\text{(triangle ineq.)}\quad &\leq \|\tilde{\Phi}_n(\tilde{y}_{n-1}) - \Phi_n(\tilde{y}_{n-1})\| + \|\Phi_n(\tilde{y}_{n-1}) - \Phi_n(y_{n-1})\|\\
\text{($\Phi$ Lipschitz)}\quad &\leq \|\tilde{\Phi}_n(\tilde{y}_{n-1}) - \Phi_n(\tilde{y}_{n-1})\| + \exp\big(L\,(t_n - t_{n-1})\big)\,\|\tilde{y}_{n-1} - y_{n-1}\|\\
\text{(recursion)}\quad &\leq \sum_{k=1}^{n} \exp\big(L\,(t_n - t_k)\big)\,\|\tilde{\Phi}_k(\tilde{y}_{k-1}) - \Phi_k(\tilde{y}_{k-1})\|.
\end{aligned}$$

# Runge-Kutta Methods

**Terminology**

$\left\|\tilde{\Phi}_k(\tilde{y}_{k-1}) - \Phi_k(\tilde{y}_{k-1})\right\|$ is called the *local*, *consistency* or *truncation error* of the numerical time propagator $\tilde{\Phi}$.

**Remark**

The bound

$$\left\|\tilde{y}_n - y_n\right\| \;\leq\; \sum_{k=1}^{n} \exp\big(L\,(t_n - t_k)\big)\; \left\|\tilde{\Phi}_k(\tilde{y}_{k-1}) - \Phi_k(\tilde{y}_{k-1})\right\|$$

can be put into words as follows.

The error at the final time $t_n$ is upper-bounded by
the sum of the errors introduced in the time steps $(t_{k-1} \to t_k)_{k=1}^{n}$
multiplied by the Lipschitz constant of $\Phi(y,t)$ derived on slide 17.

# Runge-Kutta Methods

**Discussion**

The bound

$$\left\|\tilde{y}_n - y_n\right\| \;\leq\; \sum_{k=1}^{n} \exp\bigl(L\,(t_n - t_k)\bigr)\;\left\|\tilde{\Phi}_k(\tilde{y}_{k-1}) - \Phi_k(\tilde{y}_{k-1})\right\|$$

shows that

$$\max_k \left|t_k - t_{k-1}\right| \to 0 \qquad \Longrightarrow \qquad \left\|\tilde{y}_n - y_n\right\| \to 0$$

as long as

$$\left|t_k - t_{k-1}\right| \to 0 \qquad \Longrightarrow \qquad \left\|\tilde{\Phi}_k(\tilde{y}_{k-1}) - \Phi_k(\tilde{y}_{k-1})\right\| \to 0. \tag{1}$$

We have already seen on slide 21 that (1) is indeed the case. Specifically, we have seen that

$$\left\|\tilde{\Phi}(\tilde{y}_{k-1}, t) - \Phi(\tilde{y}_{k-1}, t)\right\| = O(t^2).$$

This observation immediately yields the following theorem.

# Runge-Kutta Methods

**Thm: Convergence of Euler's method**

Denote by $y(t)$ the solution to $\dot{y} = f(y)$ and by $\tilde{y}(t)$ the numerical approximation obtained using Euler's method with the equispaced temporal mesh $(t_k = \frac{k}{n}\,T)_{k=0}^{n}$.

Then,

$$\left\|\tilde{y}(T) - y(T)\right\| = O\Big(\exp(LT)\,\tfrac{T^2}{n}\Big),$$

assuming $f(y)$ is Lipschitz continuous with Lipschitz constant $L$.

*Proof.*

$$\begin{aligned}
\left\|\tilde{y}_n - y_n\right\| &\le \sum_{k=1}^{n} \exp\big(L\,(t_n - t_k)\big)\,\left\|\tilde{\Phi}_k(\tilde{y}_{k-1}) - \Phi_k(\tilde{y}_{k-1})\right\| \\
&\le n\,\exp\big(L\,(t_n - t_0)\big)\,O\Big(\big(\tfrac{T}{n}\big)^2\Big) \\
&= \exp\big(L\,(t_n - t_0)\big)\,O\big(\tfrac{T^2}{n}\big)
\end{aligned}$$

**Numerical demonstration**

See `convergence()`.

# Runge-Kutta Methods

**Remark 1**

Note that Euler's method is second-order consistent but only first-order convergent. The heuristic reason for this is that Euler's method makes $n$ errors of magnitude $O\big(n^{-2}\big)$; hence the total error is $n\,O(n^{-2}) = O(n^{-1})$.

**Remark 2**

The above convergence estimate indicates that the number of steps required to reach a fixed error tolerance

$$\big\|\tilde{y}(T) - y(T)\big\| = O\Big(\exp(LT)\,\tfrac{T^2}{n}\Big) \le \tau$$

is $n = O\big(\tau^{-1}\,T^2\,\exp(LT)\big)$.

This once again indicates that it is difficult to solve ODEs on time-scales larger than one over the Lipschitz constant.

This is demonstrated numerically in `nsteps()`.

# Runge-Kutta Methods

To be continued.