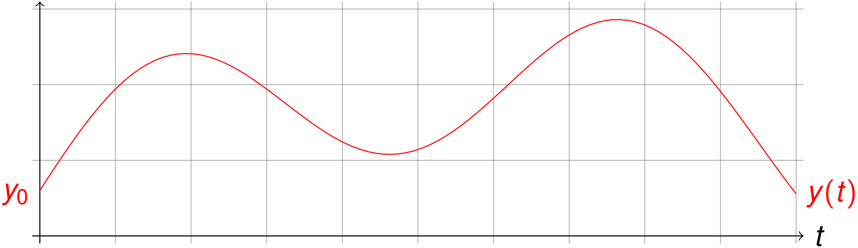
Rigorous numerical computation of polynomial differential equations over unbounded domains

Amaury Pouly
Joint work with Daniel Graça

27 November 2017

Ordinary Differential Equations (ODEs)



System of ODEs:

$$\begin{cases} y_1(0) = y_{0,1} \\ \vdots \\ y_n(0) = y_{0,n} \end{cases} \qquad \begin{cases} y_1'(t) = f_1(y_1(t), \dots, y_n(t)) \\ \vdots \\ y_n'(t) = f_n(y_1(t), \dots, y_n(t)) \end{cases}$$

More compactly:

$$y(0) = y_0 \qquad y'(t) = f(y(t))$$

Computability

Let $I = [0, a[$ and $f \in C^0(\mathbb{R}^n)$. Assume $y \in C^1(I, \mathbb{R}^d)$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)). \tag{1}$$

Can we compute $y(t) \pm 2^{-n}$ for all $t \in I$ and $n \in \mathbb{N}$?

Computability

Let $I = [0, a[$ and $f \in C^0(\mathbb{R}^n)$. Assume $y \in C^1(I, \mathbb{R}^d)$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)). \tag{1}$$

Can we compute $y(t) \pm 2^{-n}$ for all $t \in I$ and $n \in \mathbb{N}$?

- existence: Peano theorem
- uniqueness: assumption on y or f
- computability: assume f is computable

Computability

Let $I = [0, a[$ and $f \in C^0(\mathbb{R}^n)$. Assume $y \in C^1(I, \mathbb{R}^d)$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)). \tag{1}$$

Can we compute $y(t) \pm 2^{-n}$ for all $t \in I$ and $n \in \mathbb{N}$?

- existence: Peano theorem
- uniqueness: assumption on y or f
- computability: assume f is computable

Theorem (Collins and Graça)

If f is computable and (1) has a unique solution, then it is computable over its maximum interval of life I.

Computability

Let $I = [0, a[$ and $f \in C^0(\mathbb{R}^n)$. Assume $y \in C^1(I, \mathbb{R}^d)$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)). \tag{1}$$

Can we compute $y(t) \pm 2^{-n}$ for all $t \in I$ and $n \in \mathbb{N}$?

- existence: Peano theorem
- uniqueness: assumption on y or f
- computability: assume f is computable

Theorem (Collins and Graça)

If f is computable and (1) has a unique solution, then it is computable over its maximum interval of life I.

Theorem (Buescu, Campagnolo and Graça)

Computing I (or deciding if I is bounded) is undecidable, even if f is a polynomial.

Empirical approach to complexity

Assume $y : [0,1] \to \mathbb{R}^n$ satisfies $\forall t \in [0,1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous and computable.

Empirical approach to complexity

Assume $y : [0,1] \to \mathbb{R}^n$ satisfies $\forall t \in [0,1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous and computable.

- Non-rigorous: guaranteed (linear) complexity, result can be wrong
  $\to$ Unsatisfactory

Empirical approach to complexity

Assume $y : [0,1] \to \mathbb{R}^n$ satisfies $\forall t \in [0,1]$:

$$y(0) = 0, \qquad\qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous and computable.

- Non-rigorous: guaranteed (linear) complexity, result can be wrong $\to$ Unsatisfactory
- Rigorous: guaranteed result, benchmark complexity

Empirical approach to complexity

Assume $y : [0,1] \to \mathbb{R}^n$ satisfies $\forall t \in [0,1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous and computable.

- Non-rigorous: guaranteed (linear) complexity, result can be wrong $\to$ Unsatisfactory
- Rigorous: guaranteed result, benchmark complexity

Useful in practice, not that much in theory.

Nonuniform complexity-theoretic approach

Assume $y : [0, 1] \to \mathbb{R}^n$ satisfies $\forall t \in [0, 1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)).$$

Assumption on f	**Lower bound on y**	**Upper bound on y**

Nonuniform complexity-theoretic approach

Assume $y : [0,1] \to \mathbb{R}^n$ satisfies $\forall t \in [0,1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)).$$

Assumption on f	**Lower bound on y**	**Upper bound on y**
PTIME	arbitrary	computable

Nonuniform complexity-theoretic approach

Assume $y : [0,1] \to \mathbb{R}^n$ satisfies $\forall t \in [0,1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)).$$

Assumption on f	**Lower bound on y**	**Upper bound on y**
PTIME	arbitrary	computable
PTIME + Lipschitz	PSPACE-hard	PSPACE

Nonuniform complexity-theoretic approach

Assume $y : [0, 1] \to \mathbb{R}^n$ satisfies $\forall t \in [0, 1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)).$$

Assumption on f	**Lower bound on y**	**Upper bound on y**
PTIME	arbitrary	computable
PTIME + Lipschitz	PSPACE-hard	PSPACE
PTIME + C^1	PSPACE-hard	PSPACE

Nonuniform complexity-theoretic approach

Assume $y : [0,1] \to \mathbb{R}^n$ satisfies $\forall t \in [0,1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)).$$

Assumption on f	**Lower bound on y**	**Upper bound on y**
PTIME	arbitrary	computable
PTIME + Lipschitz	PSPACE-hard	PSPACE
PTIME + C^1	PSPACE-hard	PSPACE
PTIME + $C^k, k \geqslant 2$	CH-hard	PSPACE

Nonuniform complexity-theoretic approach

Assume $y : [0, 1] \to \mathbb{R}^n$ satisfies $\forall t \in [0, 1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)).$$

Assumption on f	**Lower bound on y**	**Upper bound on y**
PTIME	arbitrary	computable
PTIME + Lipschitz	PSPACE-hard	PSPACE
PTIME + C^1	PSPACE-hard	PSPACE
PTIME + $C^k, k \geqslant 2$	CH-hard	PSPACE
PTIME + analytic	—	PTIME

Nonuniform complexity-theoretic approach

Assume $y : [0,1] \to \mathbb{R}^n$ satisfies $\forall t \in [0,1]$:

$$y(0) = 0, \qquad y'(t) = f(y(t)).$$

Assumption on f	**Lower bound on y**	**Upper bound on y**
PTIME	arbitrary	computable
PTIME + Lipschitz	PSPACE-hard	PSPACE
PTIME + C^1	PSPACE-hard	PSPACE
PTIME + $C^k, k \geqslant 2$	CH-hard	PSPACE
PTIME + analytic	—	PTIME

But those results can be deceiving...

$$\begin{cases} y_1(0) = 1 \\ y_2(0) = 1 \\ \vdots \\ y_n(0) = 1 \end{cases} \qquad \begin{cases} y_1' = y_1 \\ y_2' = y_1 y_2 \\ \vdots \\ y_n' = y_{n-1} y_n \end{cases} \quad \rightarrow \quad y(t) = \mathcal{O}\left(e^{e^{\cdot^{\cdot^{\cdot^{e^t}}}}}\right)$$

y is PTIME over $[0,1]$

Nonuniform complexity: limitation

Example:

$$f \text{ PTIME analytic} \;\Rightarrow\; y \text{ PTIME} \;\Rightarrow\; y(t) \pm 2^{-n} \text{ in time } An^k$$

But:

Nonuniform complexity: limitation

Example:

$$f \text{ PTIME analytic} \;\Rightarrow\; y \text{ PTIME} \;\Rightarrow\; y(t) \pm 2^{-n} \text{ in time } An^k$$

But:

- “Hides” some of the complexity: A,k *could* be arbitrarily horrible depending on the dimension and f.

Nonuniform complexity: limitation

Example:

$$f \text{ PTIME analytic} \;\Rightarrow\; y \text{ PTIME} \;\Rightarrow\; y(t) \pm 2^{-n} \text{ in time } An^k$$

But:

- "Hides" some of the complexity: A,k *could* be arbitrarily horrible depending on the dimension and f.
- Nonconstructive: might be a different algrithm for each f, or depend on uncomputable constants.

Uniform (operator) complexity approach

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is .... Prove that $y(t) \pm 2^{-n}$ can be computed in time:

$$T(t, n, K_d, K_f)$$

where

- K_d: depends on the dimension d
- K_f: depends on f and its representation

Uniform (operator) complexity approach

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is .... Prove that $y(t) \pm 2^{-n}$ can be computed in time:

$$T(t, n, K_d, K_f)$$

where

- K_d: depends on the dimension d
- K_f: depends on f and its representation

Assumption on f	**Lower bound on T**	**Upper bound on T**

Uniform (operator) complexity approach

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is .... Prove that $y(t) \pm 2^{-n}$ can be computed in time:

$$T(t, n, K_d, K_f)$$

where

- K_d: depends on the dimension d
- K_f: depends on f and its representation

Assumption on f	**Lower bound on T**	**Upper bound on T**
computable	arbitrary	computable

Uniform (operator) complexity approach

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is .... Prove that $y(t) \pm 2^{-n}$ can be computed in time:

$$T(t, n, K_d, K_f)$$

where

- K_d: depends on the dimension d
- K_f: depends on f and its representation

Assumption on f	**Lower bound on T**	**Upper bound on T**
computable	arbitrary	computable
PTIME + analytic	arbitrary	computable

Uniform (operator) complexity approach

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is .... Prove that $y(t) \pm 2^{-n}$ can be computed in time:

$$T(t, n, K_d, K_f)$$

where

- K_d: depends on the dimension d
- K_f: depends on f and its representation

Assumption on f	**Lower bound on T**	**Upper bound on T**
computable	arbitrary	computable
PTIME + analytic	arbitrary	computable
PTIME + polynomial	arbitrary	computable

Uniform (operator) complexity approach

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is .... Prove that $y(t) \pm 2^{-n}$ can be computed in time:

$$T(t, n, K_d, K_f)$$

where

- K_d: depends on the dimension d
- K_f: depends on f and its representation

Assumption on f	**Lower bound on T**	**Upper bound on T**
computable	arbitrary	computable
PTIME + analytic	arbitrary	computable
PTIME + polynomial	arbitrary	computable
PTIME + linear	—	exponential?

Uniform (operator) complexity approach

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is .... Prove that $y(t) \pm 2^{-n}$ can be computed in time:

$$T(t, n, K_d, K_f)$$

where

- K_d: depends on the dimension d
- K_f: depends on f and its representation

Assumption on f	**Lower bound on T**	**Upper bound on T**
computable	arbitrary	computable
PTIME + analytic	arbitrary	computable
PTIME + polynomial	arbitrary	computable
PTIME + linear	—	exponential?

Problem: we cannot predict the behaviour of y based on f.

Parametrized complexity approach

Goal: Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = f(y(t)),$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is nice. Prove that $y(t) \pm 2^{-n}$ can be computed in time:

$$\text{poly}(t, n, K_d, K_f, K_y(t))$$

where

Parametrized complexity approach

Goal: Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad\qquad y'(t) = f(y(t)),$$

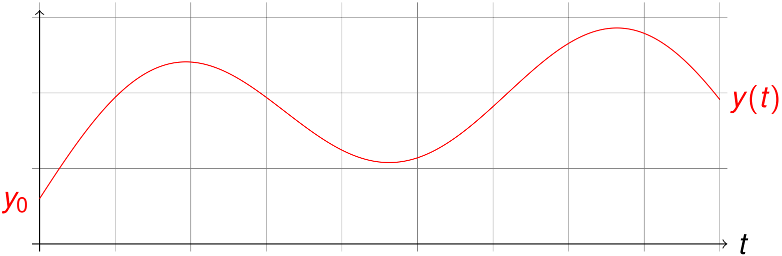
where $f : \mathbb{R}^n \to \mathbb{R}^n$ is nice. Prove that $y(t) \pm 2^{-n}$ can be computed in time:

$$\text{poly}(t, n, K_d, K_f, K_y(t))$$

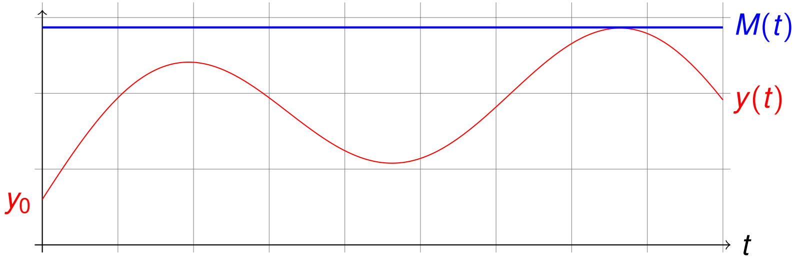
where

- K_d: depends on the dimension d
- K_f: depends on f and its representation
- K_y: is a reasonable parameter of y, ideally unknown to the algorithm (i.e. not part of the input)

Interesting parameters

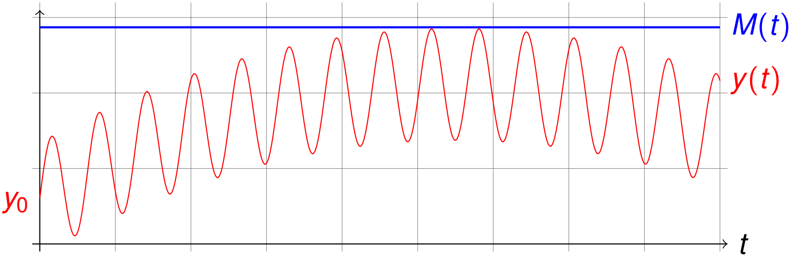


Interesting parameters



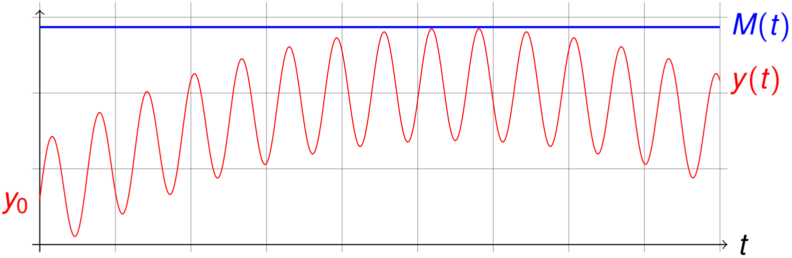
- Bounding-box: $M(t)$

Interesting parameters



- Bounding-box: $M(t)$

Interesting parameters



- Bounding-box: $M(t)$
- Length of the curve: $\int_0^t \|y'(u)\| \, du$

Parametrized complexity result

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = p(y(t)),$$

where $p : \mathbb{R}^n \to \mathbb{R}^n$ is vector of multivariate polynomials.

Theorem

Assuming $t \in I$, computing $y(t) \pm 2^{-n}$ takes time:

$$\operatorname{poly}(\deg p, \log \Sigma p, n, \ell(t_0, t))^d$$

where:

Parametrized complexity result

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = p(y(t)),$$

where $p : \mathbb{R}^n \to \mathbb{R}^n$ is vector of multivariate polynomials.

Theorem

Assuming $t \in I$, computing $y(t) \pm 2^{-n}$ takes time:

$$\operatorname{poly}(\deg p, \log \Sigma p, n, \ell(t_0, t))^d$$

where:

- Σp: sum of absolute value of coefficients of p

Parametrized complexity result

Assume $y : I \to \mathbb{R}^d$ satisfies $\forall t \in I$:

$$y(0) = 0, \qquad y'(t) = p(y(t)),$$

where $p : \mathbb{R}^n \to \mathbb{R}^n$ is vector of multivariate polynomials.

Theorem

Assuming $t \in I$, computing $y(t) \pm 2^{-n}$ takes time:

$$\operatorname{poly}(\deg p, \log \Sigma p, n, \ell(t_0, t))^d$$

where:

- Σp: sum of absolute value of coefficients of p
- $\ell(t_0, t)$: "length" of y over $[t_0, t]$

$$\ell(t_0, t) = \int_0^t \max(1, \|y'(u)\|) du$$

Note: the algorithm can find $\ell(0, t)$ automatically

Did you try to implement this algorithm?

Did you try to implement this algorithm?

Unfortunately, yes!

- “it works on simple examples”,
- don’t use it, it is too slow.

Did you try to implement this algorithm?

Unfortunately, yes!

- “it works on simple examples”,
- don’t use it, it is too slow.

Problems of this approach:

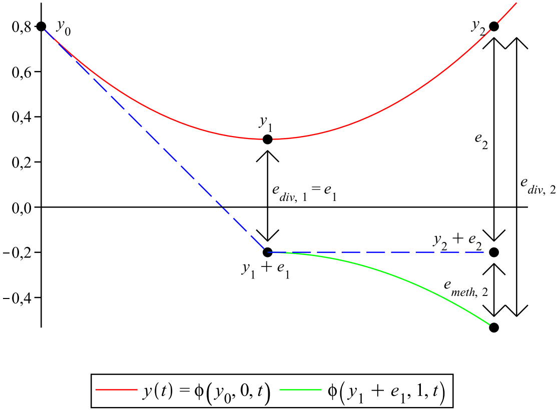
- managing complexity and correctness makes the proof complicated
- correctness comes from paper proof
- thus it is probably not *entirely* correct
- still too slow in practice

Euler method

$$y(0) = 0 \qquad y'(t) = p(y(t)) \quad t \in I$$

Time step h, discretize compute $\tilde{y}^i \approx y(ih)$:

$$y(t+h) \approx y(t) + hy'(t) \quad \rightsquigarrow \quad \tilde{y}^{i+1} = \tilde{y}^i + hp(\tilde{y}^i)$$



Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}} \varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|) du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}} \varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$
Taylor4 ($\omega = 5$)	4	$\mathcal{O}\left(\frac{L^{3/2}}{\sqrt[4]{\varepsilon}}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|) du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$
Taylor4 ($\omega = 5$)	4	$\mathcal{O}\left(\frac{L^{3/2}}{\sqrt[4]{\varepsilon}}\right)$
Smart $\left(\omega = 1 + \log \frac{L}{\varepsilon}\right)$	$\log \frac{L}{\varepsilon}$	$\mathcal{O}\left(L^{\sim 1}\right)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|)du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}} \varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$
Taylor4 ($\omega = 5$)	4	$\mathcal{O}\left(\frac{L^{3/2}}{\sqrt[4]{\varepsilon}}\right)$
Smart $\left(\omega = 1 + \log \frac{L}{\varepsilon}\right)$	$\log \frac{L}{\varepsilon}$	$\mathcal{O}\left(L^{\sim 1}\right)$
Taylor∞ ($\omega = \infty$)	∞	$\mathcal{O}(L)$

where $$L \approx \int_0^t \max(1, \|y'(u)\|) du$$

Interesting (practical ?) consequences

Compute $y(t) \pm \varepsilon$

Method	Max. Order	Number of steps
Fixed ω	$\omega - 1$	$\mathcal{O}\left(L^{\frac{\omega+1}{\omega-1}}\varepsilon^{-\frac{1}{\omega-1}}\right)$
Euler ($\omega = 2$)	1	$\mathcal{O}\left(\frac{L^3}{\varepsilon}\right)$
Taylor2 ($\omega = 3$)	2	$\mathcal{O}\left(\frac{L^2}{\sqrt{\varepsilon}}\right)$
Taylor4 ($\omega = 5$)	4	$\mathcal{O}\left(\frac{L^{3/2}}{\sqrt[4]{\varepsilon}}\right)$
Smart $\left(\omega = 1 + \log \frac{L}{\varepsilon}\right)$	$\log \frac{L}{\varepsilon}$	$\mathcal{O}\left(L^{\sim 1}\right)$
Taylor∞ ($\omega = \infty$)	∞	$\mathcal{O}(L)$
Variable	$\mathcal{O}\left(\log \frac{L}{\varepsilon}\right)$	$\mathcal{O}(L)$

where $\quad L \approx \int_0^t \max(1, \|y'(u)\|)du$

Conclusion

Solving Ordinary Differential Equations numerically:

- vastly different algorithms/results for vastly different expectations
- nonuniform complexity: imprecise/misleading
- uniform worst-case complexity: everything is hard
- uniform parametrized complexity: encouraging

Questions:

- how far can we push parametrized complexity?
- can theory bring insight to practice?

Taylor method

$$y(0) = 0 \qquad y'(t) = p(y(t)) \quad t \in I$$

Lemma: $y^{(k)}(t) = P_k(y(t)) = \text{poly}(y(t))$

Taylor method

$$y(0) = 0 \qquad y'(t) = p(y(t)) \quad t \in I$$

Lemma: $y^{(k)}(t) = P_k(y(t)) = \mathrm{poly}(y(t))$

Order K, time step h, discretize compute $\tilde{y}^i \approx y(ih)$:

$$y(t+h) \approx \sum_{j=0}^{K} \frac{h^j}{j!} y^{(j)}(t) \quad \rightsquigarrow \quad \tilde{y}^{i+1} = \sum_{j=0}^{K} \frac{h^j}{j!} P_k(\tilde{y}^i)$$

Taylor method

$$y(0) = 0 \qquad y'(t) = p(y(t)) \quad t \in I$$

Lemma: $y^{(k)}(t) = P_k(y(t)) = \text{poly}(y(t))$

Order K, time step h, discretize compute $\tilde{y}^i \approx y(ih)$:

$$y(t+h) \approx \sum_{j=0}^{K} \frac{h^j}{j!} y^{(j)}(t) \quad \rightsquigarrow \quad \tilde{y}^{i+1} = \sum_{j=0}^{K} \frac{h^j}{j!} P_k(\tilde{y}^i)$$

- **Fixed order** K: theoretically not enough

Taylor method

$$y(0) = 0 \qquad y'(t) = p(y(t)) \quad t \in I$$

Lemma: $y^{(k)}(t) = P_k(y(t)) = \mathrm{poly}(y(t))$

Order K, time step h, discretize compute $\tilde{y}^i \approx y(ih)$:

$$y(t+h) \approx \sum_{j=0}^{K} \frac{h^j}{j!} y^{(j)}(t) \quad \rightsquigarrow \quad \tilde{y}^{i+1} = \sum_{j=0}^{K} \frac{h^j}{j!} P_k(\tilde{y}^i)$$

- **Fixed order** K: theoretically not enough
- **Variable order** K: choose K depending on i, p, n and $\tilde{y}^i$

Taylor method

$$y(0) = 0 \qquad y'(t) = p(y(t)) \quad t \in I$$

Lemma: $y^{(k)}(t) = P_k(y(t)) = \text{poly}(y(t))$

Order K, time step h, discretize compute $\tilde{y}^i \approx y(ih)$:

$$y(t+h) \approx \sum_{j=0}^{K} \frac{h^j}{j!} y^{(j)}(t) \quad \rightsquigarrow \quad \tilde{y}^{i+1} = \sum_{j=0}^{K} \frac{h^j}{j!} P_k(\tilde{y}^i)$$

- **Fixed order** K: theoretically not enough
- **Variable order** K: choose K depending on i, p, n and $\tilde{y}^i$

What about h ?

- **Fixed** h: wasteful

Taylor method

$$y(0) = 0 \qquad y'(t) = p(y(t)) \quad t \in I$$

Lemma: $y^{(k)}(t) = P_k(y(t)) = \text{poly}(y(t))$

Order K, time step h, discretize compute $\tilde{y}^i \approx y(ih)$:

$$y(t+h) \approx \sum_{j=0}^{K} \frac{h^j}{j!} y^{(j)}(t) \quad \rightsquigarrow \quad \tilde{y}^{i+1} = \sum_{j=0}^{K} \frac{h^j}{j!} P_k(\tilde{y}^i)$$

- **Fixed order** K: theoretically not enough
- **Variable order** K: choose K depending on i, p, n and $\tilde{y}^i$

What about h ?

- **Fixed** h: wasteful
- **Adaptive** h: choose h depending on i, p, n and $\tilde{y}^i$

Choice of the parameters

Choice of h based on an effective lower bound on radius of convergence of the Taylor series:

Lemma: If $y' = p(y)$, $\alpha = \max(1, \|y_0\|)$, $k = \deg(p)$, $M = (k-1)\Sigma p \alpha^{k-1}$ then:

$$\left\| y^{(k)}(t) - P_k(y(t)) \right\| \leqslant \frac{\alpha (Mt)^k}{1 - Mt}$$

Choice of the parameters

Choice of h based on an effective lower bound on radius of convergence of the Taylor series:

Lemma: If $y' = p(y)$, $\alpha = \max(1, \|y_0\|)$, $k = \deg(p)$, $M = (k-1)\Sigma p \alpha^{k-1}$ then:

$$\left\| y^{(k)}(t) - P_k(y(t)) \right\| \leqslant \frac{\alpha (Mt)^k}{1 - Mt}$$

Choose $Mt \approx \frac{1}{2}$:

- $t \approx \frac{1}{M}$: adaptive step size
- local error $\approx (Mt)^k \approx 2^{-k}$: order gives the number of correct bits

Choice of the parameters

Choice of h based on an effective lower bound on radius of convergence of the Taylor series:

Lemma: If $y' = p(y)$, $\alpha = \max(1, \|y_0\|)$, $k = \deg(p)$, $M = (k-1)\Sigma p \alpha^{k-1}$ then:

$$\left\| y^{(k)}(t) - P_k(y(t)) \right\| \leqslant \frac{\alpha (Mt)^k}{1 - Mt}$$

Choose $Mt \approx \frac{1}{2}$:

- $t \approx \frac{1}{M}$: adaptive step size
- local error $\approx (Mt)^k \approx 2^{-k}$: order gives the number of correct bits

I spare you the analysis of the global error !