Verified Numerics to Solve Differential Equations

Bhavik Mehta

Imperial College London

May 14, 2026

Our toy problem

Let $\nu = 2.5$ and $\varphi = 0.5$. There is an analytic solution f for the initial value problem

$$f'(t) = f(t)(1 - f(t))$$
$$f(0) = \varphi$$

on $(-\nu, \nu)$,

Our toy problem

Let $\nu = 2.5$ and $\varphi = 0.5$. There is an analytic solution f for the initial value problem

$$f'(t) = f(t)(1 - f(t))$$
$$f(0) = \varphi$$

on $(-\nu, \nu)$, such that $\forall t \in [-\nu, \nu]$,

$$\left| f(t) - \sum y_n \left(\frac{t}{\nu}\right)^n \right| \leq 0.0143$$

Our toy problem

Let $\nu = 2.5$ and $\varphi = 0.5$. There is an analytic solution f for the initial value problem

$$f'(t) = f(t)(1 - f(t))$$
$$f(0) = \varphi$$

on $(-\nu, \nu)$, such that $\forall t \in [-\nu, \nu]$,

$$\left| f(t) - \sum y_n \left(\frac{t}{\nu}\right)^n \right| \leq 0.0143$$

where

$$y = \begin{pmatrix} 0.5 \\ 0.625 \\ 0 \\ -0.32552083 \\ 0 \\ 0.20345052 \\ 0 \\ -0.12867034 \\ 0 \\ 0.08147019 \\ 0 \\ -0.05159082 \\ 0 \\ 0.03267021 \\ 0 \\ -0.02068865 \\ 0 \\ 0.01310124 \\ 0 \\ -0.00829646 \\ 0 \\ 0.00525379 \end{pmatrix}$$

Table of Contents

Solution process

- Discretize the problem by picking a basis

Solution process

- Discretize the problem by picking a basis
- Reformulate as a zero-finding problem

Solution process

- Discretize the problem by picking a basis
- Reformulate as a zero-finding problem
- Approximate to a finite problem

Solution process

- Discretize the problem by picking a basis
- Reformulate as a zero-finding problem
- Approximate to a finite problem
- Compute an approximate solution to this problem

Solution process

- Discretize the problem by picking a basis
- Reformulate as a zero-finding problem
- Approximate to a finite problem
- Compute an approximate solution to this problem
- Verify inequalities to go back to the infinite problem

Pick a basis

- For this problem, we want an analytic solution with some radius of convergence $\geq \nu = 2.5$

Pick a basis

- For this problem, we want an analytic solution with some radius of convergence $\geq \nu = 2.5$
- Rescale $g(t) := f(\nu t)$

Pick a basis

- For this problem, we want an analytic solution with some radius of convergence $\geq \nu = 2.5$
- Rescale $g(t) := f(\nu t)$
- Discretize to $\ell^1_{\mathbb{N}}$, so

$$g(t) = \sum_i x_i t^i$$

Pick a basis

- For this problem, we want an analytic solution with some radius of convergence $\geq \nu = 2.5$
- Rescale $g(t) := f(\nu t)$
- Discretize to $\ell^1_{\mathbb{N}}$, so

$$g(t) = \sum_i x_i t^i$$

- IVP becomes

$$x_0 = \varphi$$
$$x_{n+1} = \frac{\nu}{n+1}(x_n - (x * x)_n)$$

Zero finding

- Defining $\Lambda : \ell^1 \to \ell^1$ by

$$(\Lambda x)_n := \begin{cases} 0 & n = 0 \\ \frac{1}{n} x_{n-1} & n \geq 1 \end{cases}$$

we can rewrite the problem as

$$x = \varphi + \nu\Lambda(x - x * x)$$

Zero finding

- Defining $\Lambda : \ell^1 \to \ell^1$ by

$$(\Lambda x)_n := \begin{cases} 0 & n = 0 \\ \frac{1}{n} x_{n-1} & n \geq 1 \end{cases}$$

we can rewrite the problem as

$$x = \varphi + \nu\Lambda(x - x * x)$$

- Then defining $F : \ell^1 \to \ell^1$ by

$$F(x) = x - \varphi - \nu\Lambda(x - x * x)$$

our goal becomes to find x such that $F(x) = 0$.

Newton-Kantorovich

Theorem

Let $\mathcal{X}$, $\mathcal{Y}$ be Banach spaces, with $\tilde{x} \in \mathcal{X}$ and $F : \mathcal{X} \to \mathcal{Y}$ a C^1 map. Let $A : \mathcal{Y} \to \mathcal{X}$ be injective linear. Let $0 < r \leq R$ and Y, Z_1, Z_2 be reals such that

$$\begin{aligned} \|A\,F(\tilde{x})\| &\leq Y, \\ \|I - A\,DF(\tilde{x})\| &\leq Z_1 \\ \|A\,(DF(x) - DF(\tilde{x}))\| &\leq Z_2 \quad \forall x \in B_R(\tilde{x}) \\ Y + Z_1 r + \tfrac{1}{2} Z_2 r^2 &\leq r \\ Z_1 + Z_2 r &< 1. \end{aligned}$$

Then there is a unique $x \in B_r(\tilde{x})$ such that $F(x) = 0$.

Newton-Kantorovich

Theorem

Let $\mathcal{X}$, $\mathcal{Y}$ be Banach spaces, with $\tilde{x} \in \mathcal{X}$ and $F : \mathcal{X} \to \mathcal{Y}$ a C^1 map. Let $A : \mathcal{Y} \to \mathcal{X}$ be injective linear. Let $0 < r \leq R$ and Y, Z_1, Z_2 be reals such that

$$\begin{aligned} \|A\,F(\tilde{x})\| &\leq Y, \\ \|I - A\,DF(\tilde{x})\| &\leq Z_1 \\ \|A\,(DF(x) - DF(\tilde{x}))\| &\leq Z_2 \quad \forall x \in B_R(\tilde{x}) \\ Y + Z_1 r + \tfrac{1}{2} Z_2 r^2 &\leq r \\ Z_1 + Z_2 r &< 1. \end{aligned}$$

Then there is a unique $x \in B_r(\tilde{x})$ such that $F(x) = 0$.

Newton-Kantorovich

Theorem

Let $\mathcal{X}$, $\mathcal{Y}$ be Banach spaces, with $\tilde{x} \in \mathcal{X}$ and $F : \mathcal{X} \to \mathcal{Y}$ a C^1 map. Let $A : \mathcal{Y} \to \mathcal{X}$ be injective linear. Let $0 < r \leq R$ and Y, Z_1, Z_2 be reals such that

$$\begin{aligned} \|A\,F(\tilde{x})\| &\leq Y, \\ \|I - A\,DF(\tilde{x})\| &\leq Z_1 \\ \|A\,(DF(x) - DF(\tilde{x}))\| &\leq Z_2 \quad \forall x \in B_R(\tilde{x}) \\ Y + Z_1 r + \tfrac{1}{2} Z_2 r^2 &\leq r \\ Z_1 + Z_2 r &< 1. \end{aligned}$$

Then there is a unique $x \in B_r(\tilde{x})$ such that $F(x) = 0$.

Newton-Kantorovich

Theorem

Let $\mathcal{X}$, $\mathcal{Y}$ be Banach spaces, with $\tilde{x} \in \mathcal{X}$ and $F : \mathcal{X} \to \mathcal{Y}$ a C^1 map. Let $A : \mathcal{Y} \to \mathcal{X}$ be injective linear. Let $0 < r \leq R$ and Y, Z_1, Z_2 be reals such that

$$\begin{aligned} \|A\,F(\tilde{x})\| &\leq Y, \\ \|I - A\,DF(\tilde{x})\| &\leq Z_1 \\ \|A\,(DF(x) - DF(\tilde{x}))\| &\leq Z_2 \quad \forall x \in B_R(\tilde{x}) \\ Y + Z_1 r + \tfrac{1}{2} Z_2 r^2 &\leq r \\ Z_1 + Z_2 r &< 1. \end{aligned}$$

Then there is a unique $x \in B_r(\tilde{x})$ such that $F(x) = 0$.

Sketch proof: write $T = I - A\,F$.

Recap

- Took a Taylor basis for "analytic on $[-2.5, 2.5]$" so we're now in ℓ_1

Recap

- Took a Taylor basis for "analytic on $[-2.5, 2.5]$" so we're now in ℓ_1
- Switched to a root-finding problem

Recap

- Took a Taylor basis for "analytic on $[-2.5, 2.5]$" so we're now in ℓ_1
- Switched to a root-finding problem
- Need to verify the inequalities

Recap

- Took a Taylor basis for "analytic on $[-2.5, 2.5]$" so we're now in ℓ_1
- Switched to a root-finding problem
- Need to verify the inequalities
- Moved from a continuous analytic problem to a discrete (but still infinite) computation

Recap

- Took a Taylor basis for "analytic on $[-2.5, 2.5]$" so we're now in ℓ_1
- Switched to a root-finding problem
- Need to verify the inequalities
- Moved from a continuous analytic problem to a discrete (but still infinite) computation

- Challenges in practice: finding the right basis (eg for BVP or periodic problem, could take Fourier basis), validating the inequalities

Finitizing

- Can find first N coordinates of $\tilde{x}$ numerically (e.g. Newton iteration)

Finitizing

- Can find first N coordinates of $\tilde{x}$ numerically (e.g. Newton iteration)
- Compute a $N \times N$ matrix, which when extended by 1 on the diagonal, forms an approximate inverse to F, call this A

Finitizing

- Can find first N coordinates of $\tilde{x}$ numerically (e.g. Newton iteration)
- Compute a $N \times N$ matrix, which when extended by 1 on the diagonal, forms an approximate inverse to F, call this A
- Use these as the assumptions to Newton-Kantorovich

Finitizing

- Can find first N coordinates of $\tilde{x}$ numerically (e.g. Newton iteration)
- Compute a $N \times N$ matrix, which when extended by 1 on the diagonal, forms an approximate inverse to F, call this A
- Use these as the assumptions to Newton-Kantorovich
- Prove the first three inequalities by combining matrix arithmetic and tail bound estimates

Finitizing

- Can find first N coordinates of $\tilde{x}$ numerically (e.g. Newton iteration)
- Compute a $N \times N$ matrix, which when extended by 1 on the diagonal, forms an approximate inverse to F, call this A
- Use these as the assumptions to Newton-Kantorovich
- Prove the first three inequalities by combining matrix arithmetic and tail bound estimates
- Use the values of Y, Z_1, Z_2 to deduce the best r

Finitizing

- Can find first N coordinates of $\tilde{x}$ numerically (e.g. Newton iteration)
- Compute a $N \times N$ matrix, which when extended by 1 on the diagonal, forms an approximate inverse to F, call this A
- Use these as the assumptions to Newton-Kantorovich
- Prove the first three inequalities by combining matrix arithmetic and tail bound estimates
- Use the values of Y, Z_1, Z_2 to deduce the best r
- Problem solved!

In practice

- Solving ODEs, PDEs, DDEs with discrete methods is an active field, and similar techniques have been used to solve a number of long-standing open problems

In practice

- Solving ODEs, PDEs, DDEs with discrete methods is an active field, and similar techniques have been used to solve a number of long-standing open problems
- Our example is just a toy expository problem (it can be done analytically!)

In practice

- Solving ODEs, PDEs, DDEs with discrete methods is an active field, and similar techniques have been used to solve a number of long-standing open problems
- Our example is just a toy expository problem (it can be done analytically!)
- But the same ideas work for harder problems

In practice

- Solving ODEs, PDEs, DDEs with discrete methods is an active field, and similar techniques have been used to solve a number of long-standing open problems
- Our example is just a toy expository problem (it can be done analytically!)
- But the same ideas work for harder problems
- Hard parts are: picking $\mathcal{X}$ and F (choice of function space and norm is crucial, many F can work); finding A

ODE solver

- We're *not* using a 'standard' ODE solver (like Runge-Kutta) and validating the numerics of that

ODE solver

- We're *not* using a 'standard' ODE solver (like Runge-Kutta) and validating the numerics of that
- This is what was done by Immler for the Lorenz attractor and Smale's problem

ODE solver

- We're *not* using a 'standard' ODE solver (like Runge-Kutta) and validating the numerics of that
- This is what was done by Immler for the Lorenz attractor and Smale's problem
- Half of the informal proof there was classical analysis (numerical methods fail)

Table of Contents

Overall

- Formalizing all the previous pieces, we have the original theorem

Overall

- Formalizing all the previous pieces, we have the original theorem

Theorem

There is an analytic solution f for the initial value problem

$$f'(t) = f(t)(1 - f(t))$$
$$f(0) = \varphi$$

on $(-\nu, \nu)$, *such that* $\forall t \in [-\nu, \nu]$,

$$\left| f(t) - \sum y_n \left(\frac{t}{\nu}\right)^n \right| \leq 0.0143$$

Overall

- Formalizing all the previous pieces, we have the original theorem

Theorem

There is an analytic solution f for the initial value problem

$$f'(t) = f(t)(1 - f(t))$$
$$f(0) = \varphi$$

on $(-\nu, \nu)$, *such that* $\forall t \in [-\nu, \nu]$,

$$\left| f(t) - \sum y_n \left(\frac{t}{\nu}\right)^n \right| \leq 0.0143$$

- We compute r, y_n, A, Y, Z_1, Z_2 'offline' (in Julia) and paste in the values

Tail bounds

- The notes we followed didn't give details of the tail bound proof for Z_1

Tail bounds

- The notes we followed didn't give details of the tail bound proof for Z_1
- We had to change the truncation limit to get a rigorous proof there (convolving two points in ℓ^1 with support $0, \ldots, 21$ produces something with support bigger than 21)

Tail bounds

- The notes we followed didn't give details of the tail bound proof for Z_1
- We had to change the truncation limit to get a rigorous proof there (convolving two points in ℓ^1 with support $0, \ldots, 21$ produces something with support bigger than 21)
- But this increased Z_1, and meant that no r satisfied the inequalities

Tail bounds

- The notes we followed didn't give details of the tail bound proof for Z_1
- We had to change the truncation limit to get a rigorous proof there (convolving two points in ℓ^1 with support $0, \ldots, 21$ produces something with support bigger than 21)
- But this increased Z_1, and meant that no r satisfied the inequalities
- So we increased N as well, and then a solution is recovered

Tail bounds

- The notes we followed didn't give details of the tail bound proof for Z_1
- We had to change the truncation limit to get a rigorous proof there (convolving two points in ℓ^1 with support $0, \ldots, 21$ produces something with support bigger than 21)
- But this increased Z_1, and meant that no r satisfied the inequalities
- So we increased N as well, and then a solution is recovered
- These calculations are very fiddly informally!

Efficiency

- Our first version took hours to compile, the current version takes 2 minutes

Efficiency

- Our first version took hours to compile, the current version takes 2 minutes
- I think it's doable in under a second, and I think I know how to get there

Efficiency

- Our first version took hours to compile, the current version takes 2 minutes
- I think it's doable in under a second, and I think I know how to get there
- Profiling is *incredibly* important - Lean's performance is hard to predict

Efficiency

- Our first version took hours to compile, the current version takes 2 minutes
- I think it's doable in under a second, and I think I know how to get there
- Profiling is *incredibly* important - Lean's performance is hard to predict
- Avoiding grind made some things faster (it spent loads of time in pointless typeclass search)

Efficiency example

- If you write an explicit vector using `![1, 2, 3]` notation (i.e. `Fin N` $\to$ $\mathbb{R}$), then evaluating its entries takes *exponential* time in the entry

Efficiency example

- If you write an explicit vector using `![1, 2, 3]` notation (i.e. `Fin N` $\to$ $\mathbb{R}$), then evaluating its entries takes *exponential* time in the entry
- So taking the 21th entry is twice as slow as taking the 20th *in the compiler*, using `#eval`

Efficiency example

- If you write an explicit vector using `![1, 2, 3]` notation (i.e. `Fin N` $\to$ $\mathbb{R}$), then evaluating its entries takes *exponential* time in the entry
- So taking the 21th entry is twice as slow as taking the 20th *in the compiler*, using `#eval`
- (demo)

Profiling

Three things to profile:

- Elaborator: Use `set_option trace.profiler true` (with trace nodes) or `set_option profiler true`

Profiling

Three things to profile:

- Elaborator: Use `set_option trace.profiler true` (with trace nodes) or `set_option profiler true`
- Kernel: Use `set_option trace.profiler true` to get a number, and `set_option diagnostics true` to get some understanding (neither of these are perfect)

Profiling

Three things to profile:

- Elaborator: Use `set_option trace.profiler true` (with trace nodes) or `set_option profiler true`
- Kernel: Use `set_option trace.profiler true` to get a number, and `set_option diagnostics true` to get some understanding (neither of these are perfect)
- Compiled code: Use a C profiler like `samply record lake build`

Calculations

- Lean 4 is very fast at calculations on natural numbers, so if you can reduce to that, it's very fast

Calculations

- Lean 4 is very fast at calculations on natural numbers, so if you can reduce to that, it's very fast
- But the kernel is not optimised (so that it can be small and trusted)

Calculations

- Lean 4 is very fast at calculations on natural numbers, so if you can reduce to that, it's very fast
- But the kernel is not optimised (so that it can be small and trusted)
- So you need to help it along the way (demo, if time)

Usability

- We'd like this to be a framework for proofs in this area to be cleanly verified

Usability

- We'd like this to be a framework for proofs in this area to be cleanly verified
- The *entire* example is under 400 lines (demo)

Usability

- We'd like this to be a framework for proofs in this area to be cleanly verified
- The *entire* example is under 400 lines (demo)
- But the API is still in flux; more examples should help settle it

Usability

- We'd like this to be a framework for proofs in this area to be cleanly verified
- The *entire* example is under 400 lines (demo)
- But the API is still in flux; more examples should help settle it
- Want built-in support for other bases and other spaces (e.g. Sobolev spaces)

Usability

- We'd like this to be a framework for proofs in this area to be cleanly verified
- The *entire* example is under 400 lines (demo)
- But the API is still in flux; more examples should help settle it
- Want built-in support for other bases and other spaces (e.g. Sobolev spaces)
- Want to support other techniques, not just Newton-Kantorovich

Table of Contents

Next steps

- Optimise it further

Next steps

- Optimise it further
- Solve periodic problems (by picking other bases)

Next steps

- Optimise it further
- Solve periodic problems (by picking other bases)
- Solve certain boundary value problems (by picking other bases)

Next steps

- Optimise it further
- Solve periodic problems (by picking other bases)
- Solve certain boundary value problems (by picking other bases)
- Solve certain PDEs

Next steps

- Optimise it further
- Solve periodic problems (by picking other bases)
- Solve certain boundary value problems (by picking other bases)
- Solve certain PDEs
- Avoid using Julia?

Summary

- Numeric calculations can lead to rigorous outputs

Summary

- Numeric calculations can lead to rigorous outputs
- Discrete computations can solve continuous problems

Summary

- Numeric calculations can lead to rigorous outputs
- Discrete computations can solve continuous problems
- Rigorous numerics for differential equations are now more rigorous

Summary

- Numeric calculations can lead to rigorous outputs
- Discrete computations can solve continuous problems
- Rigorous numerics for differential equations are now more rigorous
- Efficiency and usability can be subtle

Summary

- Numeric calculations can lead to rigorous outputs
- Discrete computations can solve continuous problems
- Rigorous numerics for differential equations are now more rigorous
- Efficiency and usability can be subtle
- This is just the beginning: it will get faster, it will get more ergonomic, and we will solve harder problems

Thank you for your attention!