Numerically verified proofs in pure maths

Daniel Platt (Imperial College London)
Accelerate Programme Lunchtime Seminar, University of Cambridge, 20 Apr 2026

What's a numerically verified proof? In pure maths we want to prove theorems, usually using pen and paper. On the other side there exist hundreds of very elaborate ways to approximately solve equations, for example physics-informed neural networks. Due to the advent of greater computational power it has recently become possible to use such approximate solutions in a theorem proof. In the talk, I'll explain how that works in a toy example and then briefly mention some applications of this in pure maths.

Mathematical proof versus numerics

Question: does the equation $x^2 - 2 = 0$ have a solution?

- Pen and paper proof:
- $x^2 - 2 = 0 \iff x^2 = 2$
- Taking square roots, this is equivalent to $x = \pm\sqrt{2}$.
- $\Rightarrow$ a solution exists.

- Numerics:

```
from scipy.optimize import fsolve

x, = fsolve(lambda x: x**2 - 2, x0=1.0)

print(f"x = {x}")
# 1.4142135623730947

print(f"residual x^2 - 2 = {x**2 - 2}")
# -8.881784197001252e-16
```

Can one use numerics for giving a proof?
Yes! Numerically verified proofs

Mathematical proof versus numerics

Question: does the equation $x^2 - 2 = 0$ have a solution?

- Pen and paper proof:
- $x^2 - 2 = 0 \iff x^2 = 2$
- Taking square roots, this is equivalent to $x = \pm\sqrt{2}$.
- $\Rightarrow$ a solution exists.

- Numerics:

```
from scipy.optimize import fsolve

x, = fsolve(lambda x: x**2 - 2, x0=1.0)

print(f"x = {x}")
# 1.4142135623730947

print(f"residual x^2 - 2 = {x**2 - 2}")
# -8.881784197001252e-16
```

Can one use numerics for giving a proof?
Yes! Numerically verified proofs

Mathematical proof versus numerics

Question: does the equation $x^2 - 2 = 0$ have a solution?

- Pen and paper proof:
- $x^2 - 2 = 0 \iff x^2 = 2$
- Taking square roots, this is equivalent to $x = \pm\sqrt{2}$.
- $\Rightarrow$ a solution exists.

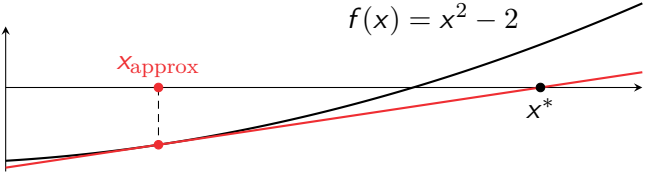
- Numerics:

```
from scipy.optimize import fsolve

x, = fsolve(lambda x: x**2 - 2, x0=1.0)

print(f"x = {x}")
# 1.4142135623730947

print(f"residual x^2 - 2 = {x**2 - 2}")
# -8.881784197001252e-16
```

Can one use numerics for giving a proof?
Yes! Numerically verified proofs

Newton method and Newton-Kantorovich

- Newton method: start with approximate solution x_{approx} and improve iteratively



Theorem (Newton-Kantorovich theorem)

Assume $|f''(x)| \leq M$ for all $x \in \mathbb{R}$ and assume

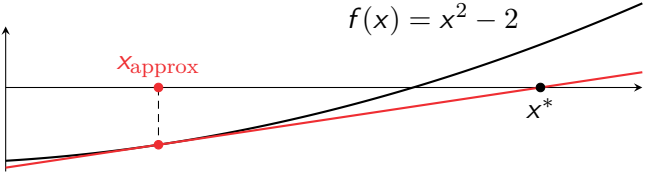
$$2M|f(x_{approx})| < |f'(x_{approx})|^2.$$

Then there exists $\hat{x}$ such that $f(\hat{x}) = 0$.

- Apply to our case:
  - $f''(x) = 2$, so $M = 2$
  - $x_{\text{approx}} = 1.41$, so $|f(x_{\text{approx}})| = 0.0119$
  - $f'(x) = 2x$, so $|f'(x_{\text{approx}})|^2 = 7.9524$
- $2 \cdot 2 \cdot 0.0119 < 7.9524$ is true!
  $\Rightarrow$ solution exists

Newton method and Newton-Kantorovich

- Newton method: start with approximate solution x_{approx} and improve iteratively



Theorem (Newton-Kantorovich theorem)

Assume $|f''(x)| \leq M$ *for all* $x \in \mathbb{R}$ *and assume*

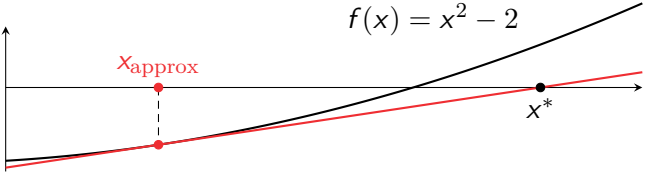
$$2M|f(x_{approx})| < |f'(x_{approx})|^2.$$

Then there exists $\hat{x}$ *such that* $f(\hat{x}) = 0$.

- Apply to our case:
  - $f''(x) = 2$, so $M = 2$
  - $x_{\text{approx}} = 1.41$, so $|f(x_{\text{approx}})| = 0.0119$
  - $f'(x) = 2x$, so $|f'(x_{\text{approx}})|^2 = 7.9524$
- $2 \cdot 2 \cdot 0.0119 < 7.9524$ is true!
  $\Rightarrow$ solution exists

Newton method and Newton-Kantorovich

- Newton method: start with approximate solution x_{approx} and improve iteratively



Theorem (Newton-Kantorovich theorem)

Assume $|f''(x)| \leq M$ *for all* $x \in \mathbb{R}$ *and assume*

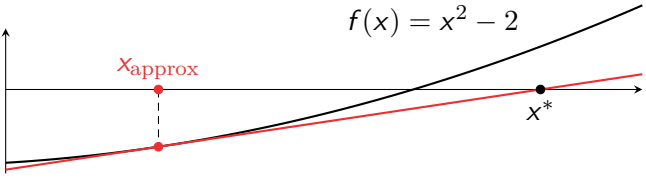
$$2M|f(x_{approx})| < |f'(x_{approx})|^2.$$

Then there exists $\hat{x}$ *such that* $f(\hat{x}) = 0$.

- Apply to our case:
  - $f''(x) = 2$, so $M = 2$
  - $x_{\text{approx}} = 1.41$, so $|f(x_{\text{approx}})| = 0.0119$
  - $f'(x) = 2x$, so $|f'(x_{\text{approx}})|^2 = 7.9524$
- $2 \cdot 2 \cdot 0.0119 < 7.9524$ is true!
  $\Rightarrow$ solution exists

Newton method and Newton-Kantorovich

- Newton method: start with approximate solution x_{approx} and improve iteratively



Theorem (Newton-Kantorovich theorem)

Assume $|f''(x)| \leq M$ *for all* $x \in \mathbb{R}$ *and assume*

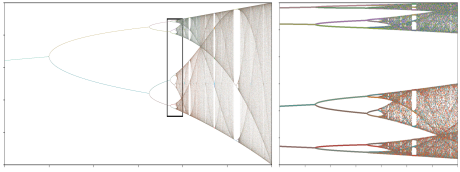
$$2M|f(x_{approx})| < |f'(x_{approx})|^2.$$

Then there exists $\hat{x}$ *such that* $f(\hat{x}) = 0$.

- Apply to our case:
  - $f''(x) = 2$, so $M = 2$
  - $x_{\text{approx}} = 1.41$, so $|f(x_{\text{approx}})| = 0.0119$
  - $f'(x) = 2x$, so $|f'(x_{\text{approx}})|^2 = 7.9524$
- $2 \cdot 2 \cdot 0.0119 < 7.9524$ is true!
  $\Rightarrow$ solution exists

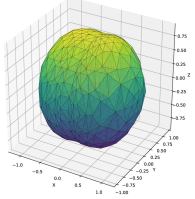
Applications, applications, applications

- Apply to differential equations
- Example 1: Feigenbaum conjectures [Lan82]

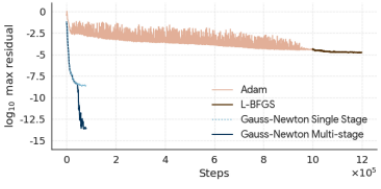


- Example 2: Sphere curvature [Pla26]

(img "A Machine Learning Approach to the Nirenberg Problem")

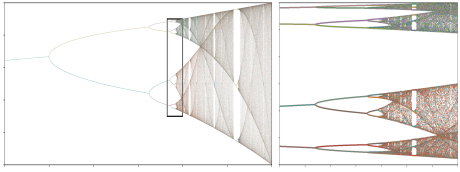


- Example 3: Fluids+PINNs [WBM+25]



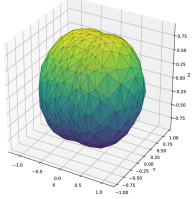
Applications, applications, applications

- Apply to differential equations
- Example 1: Feigenbaum conjectures [Lan82]

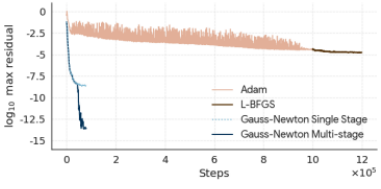


- Example 2: Sphere curvature [Pla26]

(img "A Machine Learning Approach to the Nirenberg Problem")

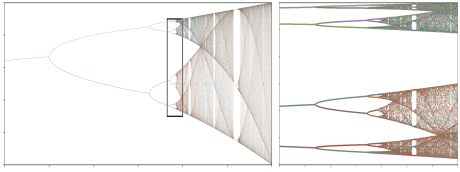


- Example 3: Fluids+PINNs [WBM+25]



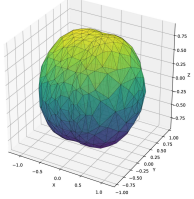
Applications, applications, applications

- Apply to differential equations
- Example 1: Feigenbaum conjectures [Lan82]

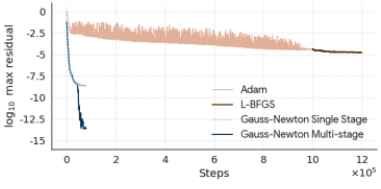


- Example 2: Sphere curvature [Pla26]

(img "A Machine Learning Approach to the Nirenberg Problem")

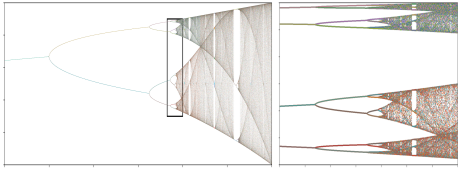


- Example 3: Fluids+PINNs [WBM+25]



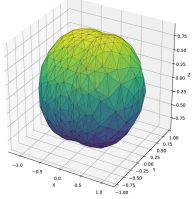
Applications, applications, applications

- Apply to differential equations
- Example 1: Feigenbaum conjectures [Lan82]

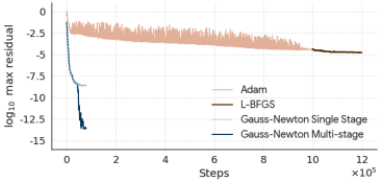


- Example 2: Sphere curvature [Pla26]

(img "A Machine Learning Approach to the Nirenberg Problem")



- Example 3: Fluids+PINNs [WBM+25]



Thank you for the attention!

References I


Oscar E Lanford.
A computer-assisted proof of the feigenbaum conjectures.
1982.


Daniel Platt.
Non-uniqueness and symmetries for the nirenberg problem using computer assistance, 2026.


Yongji Wang, Mehdi Bennani, James Martens, Sébastien Racanière, Sam Blackwell, Alex Matthews, Stanislav Nikolov, Gonzalo Cao-Labora, Daniel S. Park, Martin Arjovsky, Daniel Worrall, Chongli Qin, Ferran Alet, Borislav Kozlovskii, Nenad Tomašev, Alex Davies, Pushmeet Kohli, Tristan Buckmaster, Bogdan Georgiev, Javier Gómez-Serrano, Ray Jiang, and Ching-Yao Lai.
Discovery of unstable singularities, 2025.