

Inverse Yogiisms

*Lloyd N. Trefethen*¹

1 Berra backwards

The great New York Yankees catcher Yogi Berra died in September, 2015. Berra was famous for his quirky sayings, like these:

“It ain’t over till it’s over.”

“When you come to a fork in the road, take it.”

“It gets late early out there.”

“A nickel ain’t worth a dime anymore.”

“I always thought that record would stand until it was broken.”

“You wouldn’t have won if we’d beaten you.”

“Nobody goes there anymore, it’s too crowded.”

Maybe Berra never said half the things he said, but that’s not the point. We have here a brand of malapropisms that people have been enjoying for years.

It’s pretty easy to spot the trick that animates these quips. Yogiisms are statements that, if taken literally, are meaningless or contradictory or nonsensical or tautological — yet nevertheless convey something true. It’s a clever twist that gets us smiling and paying attention. If you like, you could argue that literature and art sometimes use the same device. A Yogiism is like a Picasso painting, you could say, messing with reality in a manner that catches our interest and still conveys a truth.

But I want to stay with words and their meanings. I think Yogiisms hold a special lesson for mathematicians, because our characteristic pitfall, I propose, is the *inverse Yogiism*: the statement that is literally true, yet conveys something false.

At some level, we’re all well aware that saying useless true things is an occupational hazard. Just think of that joke about the people lost in a hot air balloon who shout “Where are we?” to a man on the ground. “You’re in a balloon!” the mathematician answers. (I have heard this joke far too often.)

So we all know in a general way about our habit of taking things literally. My proposal is that this phenomenon is more important than we may realize, and the notion of an inverse Yogiism can help us focus on it. Inverse Yogiisms in mathematics, and in science more generally, can impede progress sometimes for generations. I will describe two examples from my own career and then mention a third topic, more open-ended, that may be a very big example indeed.

¹Nick Trefethen is Professor of Numerical Analysis at Oxford University. His email address is `trefethen@maths.ox.ac.uk`.

2 Faber's theorem on polynomial interpolation

The early 1900s was an exciting time for the constructive theory of real functions. The old idea of a function as given by a formula had broadened to include arbitrary mappings defined pointwise, and connecting the two notions was a matter of wide interest. In particular, mathematicians were concerned with the problem of approximating a continuous function f defined on an interval such as $[-1, 1]$ by a polynomial p . Weierstrass's theorem of 1885 had shown that arbitrarily close approximations always exist, and by 1912, alternative proofs had been published by Picard, Lerch, Volterra, Lebesgue, Mittag-Leffler, Fejér, Landau, de la Vallée Poussin, Jackson, Sierpiński, and Bernstein.

How could polynomial approximations be constructed? The simplest method would be interpolation by a degree- n polynomial in a set of $n + 1$ distinct points in $[-1, 1]$. Runge showed in 1900 that interpolants in equally spaced points will not generally converge to f as $n \rightarrow \infty$, even for analytic f . On the other hand, Chebyshev grids with their points clustered near ± 1 do much better. Yet around 1912, it became clear to Bernstein, Jackson, and Faber that no system of interpolation points could work for all functions. The famous result was published by Faber in 1914, and here it is in his words and notation (translated from [4]).

Faber's Theorem. *There does not exist any set E of interpolation points $x_i^{(n)}$ in $s = (-1, 1)$ ($n = 1, 2, \dots$; $i = 1, 2, \dots, n + 1$) with the property that every continuous function $\Phi(x)$ in s can be represented as the uniform limit of the degree- n polynomials taking the same values as Φ for $x = x_i^{(n)}$.*

The proof nowadays (though not yet in 1914) makes elegant use of the uniform boundedness principle.

Faber's theorem is true, and moreover, it is beautiful. Let me now explain how its influence has been unfortunate.

The field of numerical analysis took off as soon as computers were invented, and the approximation of functions was important in every area. You might think that polynomial interpolation would have been one of the standard tools from the start, and to some extent this was the case. However, practitioners must have often run into trouble when they worked with polynomial interpolants — usually because of using equispaced points or unstable algorithms, I suspect — and Faber's theorem must have looked like some kind of explanation of what was going wrong. The hundreds of textbooks that soon began to be published fell into the habit of teaching students that interpolation is a dangerous technique, not to be trusted. Here are some illustrations.

Isaacson and Keller, *Analysis of Numerical Methods* (1966), p. 275:

It is not generally true that higher degree interpolation polynomials yield more accurate approximations.

Kahaner, Moler, and Nash, *Numerical Methods and Software* (1989), p. 94:

Polynomial interpolants rarely converge to a general continuous function.

Kincaid and Cheney, *Numerical Analysis* (1991), p. 319:

The surprising state of affairs is that for most continuous functions, the quantity $\|f - p_n\|_\infty$ will not converge to 0.

Stoer and Bulirsch, *Introduction to Numerical Analysis* (1993), p. 51:

It should not be assumed that finer and finer samplings of the function f will lead to better and better approximations through interpolation.

Stewart, *Afternotes on Numerical Analysis* (1996), p. 153:

Unfortunately, there are functions for which interpolation at the Chebyshev points fails to converge.

Gautschi, *Numerical Analysis: An Introduction* (1997), p. 79:

It is not possible, therefore, to conclude... that Lagrange interpolation converges uniformly on $[a, b]$ for any continuous function, not even for judiciously selected nodes; indeed, one knows that it does not.

Quarteroni, Sacco, and Saleri, *Numerical Mathematics* (2000), p. 331:

Thus, polynomial interpolation does not allow for approximating *any* continuous function....

What a load of inverse Yogiisms! Statements like these, which appear in so many of the textbooks, give entirely the wrong impression. In fact, polynomial interpolation in Chebyshev points is a powerful and reliable method for approximation of functions. The Chebfun software system routinely works with degrees in the thousands [2].

The flaw in the logic is that Faber's theorem says nothing if f is smooth [7]. If f is Lipschitz continuous, that is more than enough to guarantee convergence of interpolants in Chebyshev points, and if it has a k th derivative of bounded variation, the error in the degree- n interpolant is of size $O(n^{-k})$. If f is analytic, the convergence is at a geometric rate $O(\rho^{-n})$, $\rho > 1$. Moreover, there are methods for computing these interpolants that are fast and numerically stable, notably the so-called barycentric interpolation formula.

So the idea that polynomial interpolants can't be trusted is a myth: a myth that has drawn strength from an impeccable theorem. Make sure your functions are Lipschitz continuous or better, as is easily done in almost any application, and Faber's theorem ceases to be applicable. In fact, polynomial interpolation in Chebyshev points has the same power and robustness as discrete Fourier analysis, to which it is essentially equivalent. We must hope that the numerical analysis textbooks of future generations will begin to tell students this.

3 Squire's theorem on hydrodynamic instability

We now move from numerical analysis to one of the oldest problems of fluid mechanics. Consider the idealized *plane Poiseuille flow* of a Newtonian liquid or gas in an infinite channel between two flat plates. (The mathematics is similar for other geometries such as a circular pipe, as investigated by Reynolds in 1883.) The flow is governed by the Navier-Stokes equations, and the key parameter is the Reynolds number Re , a nondimensionalized velocity.

Will the flow be laminar, or turbulent? At low values of Re , it is the laminar solution one sees in the laboratory, a smooth parallel downstream flow with a fixed velocity profile in the shape of a parabola. At high Re , though the laminar flow remains a mathematically valid solution of the equations, what one sees in the lab are the chaotic whirls and eddies of turbulence. Now if this is so, it seems clear that for high Re , the laminar flow must be unstable in the sense

that small perturbations of that flow may get amplified. In an analysis going back to Orr and Sommerfeld in 1907–08, one makes this precise by linearizing the equations about the laminar solution, obtaining a linear operator $\mathcal{L}_{\text{Re}}$ that governs the evolution of infinitesimal perturbations. If $\mathcal{L}_{\text{Re}}$ has an eigenfunction corresponding to an eigenvalue in the right half of the complex plane, this represents an infinitesimal perturbation that can grow exponentially, so the flow should be unstable; and if not, it should be stable.

This brings us to the elegant result published by Herbert Brian Squire in 1933. The geometry of our planar domain is 3D, with variables x (streamwise), y (perpendicular to the plates), and z (spanwise). Analyzing the linearized operator for this 3D flow is going to be complicated. Squire’s theorem, however, tells us we can ignore the z direction and just do a 2D analysis. Here is the statement from his original paper [6].

Squire’s Theorem. *Any instability which may be present for three-dimensional disturbances is also present for two-dimensional disturbances at a lower value of Reynolds’ number.*

The influence of Squire’s theorem can be seen all across the literature of mathematical fluid mechanics. Whenever you see an analysis involving the famous Orr–Sommerfeld equation, the authors have probably taken a 3D flow problem and reduced it to 2D. For the plane Poiseuille configuration, the theory tells us that the 2D instability sets in at a critical Reynolds number $\text{Re}_c \approx 5772.22$, a threshold first calculated accurately by Orszag. For $\text{Re} < \text{Re}_c$ we should expect stability and laminar flow, and for $\text{Re} > \text{Re}_c$, instability and turbulence.

Here are summaries from some books.

Lin, *The Theory of Hydrodynamic Stability* (1967), p. 27:

We shall now show, following Squire (1933), that the problem of three-dimensional disturbances is actually equivalent to a two-dimensional problem at a *lower* Reynolds number.

Tritton, *Physical Fluid Dynamics* (1977), p. 220:

...there is a result, known as Squire’s theorem, that in linear stability theory the critical Reynolds number for a two-dimensional parallel flow is lowest for two dimensional perturbations. We may thus restrict attention to these.

Drazin and Reid, *Hydrodynamic Stability* (1981), p. 155:

Squire’s theorem. To obtain the minimum critical Reynolds number it is sufficient to consider only two-dimensional disturbances.

Friedlander and Serre, eds., *Handbook of Mathematical Fluid Dynamics*, v. 3 (2002), p. 248:

THEOREM 1.1 (Squire’s theorem, 1933). To each unstable three-dimensional disturbance there corresponds a more unstable two-dimensional one.

Sengupta, *Instabilities of Flows and Transition to Turbulence* (2012), p. 82:

In a two-dimensional boundary layer with real wave numbers, instability appears first for two-dimensional disturbances.

These and other sources are in agreement on a very clear picture, and only one thing is amiss: the picture is wrong! In the laboratory, observed structures related to transition to turbulence are almost invariably three-dimensional. Moreover, it is difficult to spot any change of flow behavior at $\text{Re} \approx 5772.22$. For $\text{Re} < \text{Re}_c$, many flows are turbulent when we expect them to be laminar. For $\text{Re} > \text{Re}_c$, many flows are laminar when we expect them to be turbulent. What is going on?

The flaw in the logic is that eigenmodal analysis applies in the limit $t \rightarrow \infty$, whereas the values of t achievable in the laboratory rarely exceed 100. (Thanks to the nondimensionalization, t is related to the *length* of a flow apparatus relative to its width.) Consequently, high-Reynolds number flows normally do not become turbulent in an eigenmodal fashion [8]. On the one hand, the exponential growth rates of unstable eigenfunctions, known as Tollmien–Schlichting waves, are typically so low that in a laboratory setup they struggle to amplify a perturbation by even a factor of 2. This is why laminar flow is often observed with $\text{Re} \gg \text{Re}_c$. On the other hand, much faster transient amplification mechanisms are present for 3D perturbations, even for $\text{Re} \ll \text{Re}_c$. The perturbations involved are not eigenfunctions, and in principle they would die out as $t \rightarrow \infty$ if they started out truly infinitesimal: Squire’s theorem is, of course, literally true. But the transient growth of 3D perturbations is so substantial that in a real flow, small finite disturbances may quickly be raised to a level where nonlinearities kick in. In assuring us that the most dangerous disturbances are two-dimensional, Squire’s theorem has told us exactly the wrong place to look for hydrodynamic instability.

4 P = NP?

The most famous problem in computer science, which is also one of the million-dollar Clay Millennium Prize Problems, is the celebrated question “ $P = NP$?” This puzzle remains unresolved half a century after it was first posed by Cook and Levin in 1971.

Some computational problems can be solved by fast algorithms, and others only by slow ones. One might expect a continuum of difficulty, but the unlocking observation was that there is a gulf between *polynomial time* and *exponential time* algorithms. Inverting an $n \times n$ matrix, say, can be done in $O(n^3)$ operations or less, so we deal routinely with matrices with dimensions in the thousands. Finding the shortest route for a salesman visiting n cities, on the other hand, requires C^n operations for some $C > 1$ by all algorithms yet discovered. As $n \rightarrow \infty$, the difference between n^C and C^n looks like a clean binary distinction. And there is a great class of thousands of problems, the NP-complete problems, that have been proved to be equivalent in the sense that all of them can be solved in polynomial time or none of them can — and nobody knows which.

It’s an extraordinary gap in our knowledge. If I may pick two mathematical mysteries that I hope will be resolved before I die, they are the Riemann hypothesis and “ $P=NP$?” It is so crisp, and so important!

Yet Yogi Berra seems to be looking over our shoulders. Computers are

millions of times more powerful than they were in 1971, increasing the tractable size of n in every problem known to man, so one might expect that the gulf between P and the best known algorithms for NP, which seemed significant already in 1971, should have opened up by now to a canyon so deep we can't see its bottom. Yet in the event, nothing so straightforward has happened. Some NP-complete problems still defeat us. Others are easily solvable in many instances. For example, one of the classic NP-complete problems is "SAT," involving the satisfiability of Boolean expressions. SAT solvers have become so powerful that they are now a standard computational tool, solving problem instances of scales in the thousands and even millions [5]. Surprisingly powerful methods have been developed for other NP-complete problems too, including integer programming [1] and the traveling salesman problem itself [3].

So is there a logical flaw in " $P=NP?$ ", as with Faber's theorem and Squire's theorem? I would not go so far as to say this, but it is certainly the case that, once again, the precision of a mathematical formulation has encouraged us to think the truth is simpler than it is. A typical NP-complete problem measures complexity by the *worst case time* required to deliver the *optimal solution*. Experience has shown that in practice, both ends of this formulation are negotiable. For some NP-complete problems, like SAT, the worst case indeed looks exponential but in practice it is rare for a problem instance to come close to the worst case. For others, like the "max-cut" problem, it can be proved that even in the worst case one can solve the problem in polynomial time, if one is willing to miss the optimum by a few percent (for max-cut, 13% is enough). A field of approximation algorithms has grown up that develops algorithms of this flavor [9]. Often these algorithms rely on tools of continuous mathematics to approximate problems formulated discretely. Indeed, the whole basis of " $P=NP?$ " is a discrete view of the world, and the distracting sparkle of this great unsolved problem may have delayed the recognition that often, continuous algorithms have advantages even for discrete problems. As scientists we must always simplify the world to make sense of it; the challenge is to not get trapped by our simplifications.

5 Coda

When we say something precisely and even prove that it's true, we open ourselves to the risk of inverse Yogiisms. Would it be better if mathematicians didn't try so hard to be precise? Certainly not! Rigorous theorems are the pride of mathematics, which enable this unique subject to advance from one century to the next. The point is only that we must always strive to examine a problem from different angles, to think widely about its context as well as technically about its details. Or as Yogi put it,

"Sometimes you can see a lot just by looking."

Acknowledgments

Many people made good suggestions as I was writing this essay. I would like particularly to acknowledge Tadashi Tokieda and David Williamson.

References

Books cited in the quotations are readily tracked down from the information given there; I have quoted from the earliest editions I had access to. Other references are as below.

- [1] R. E. Bixby, A brief history of linear and mixed-integer programming computation, *Documenta Mathematica* (2012), 107–121.
- [2] Chebfun software project, www.chebfun.org.
- [3] W. J. Cook, The Traveling Salesman Problem, www.math.uwaterloo.ca/tsp/.
- [4] G. Faber, Über die interpolatorische Darstellung stetiger Funktionen, *Jber. Deutsch. Math. Verein.* 23 (1914), 192–210.
- [5] D. E. Knuth, *Satisfiability*, v. 4 fascicle 6 of *The Art of Computer Programming*, 2015.
- [6] H. B. Squire, On the stability for three-dimensional disturbances of viscous fluid flow between parallel walls, *Proc. Roy. Soc. Lond. A* 142 (1933), 621–628.
- [7] L. N. Trefethen, *Approximation Theory and Approximation Practice*, SIAM, 2013.
- [8] L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll, Hydrodynamic stability without eigenvalues, *Science* 261 (1993), 578–584.
- [9] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*, Cambridge U. Press, 2011.



Figure 1: Yogi Berra (1925–2015) in 1953. The New York Yankees beat the Brooklyn Dodgers that year to win the World Series for the 16th time.



Figure 2: Georg Faber (1877–1966) in 1909, a few years after his invention of Faber polynomials for series expansions on sets in the complex plane. Faber later moved to the Technische Hochschule in Munich, where he spent the years 1916–1946. [Figure needs to be cropped]

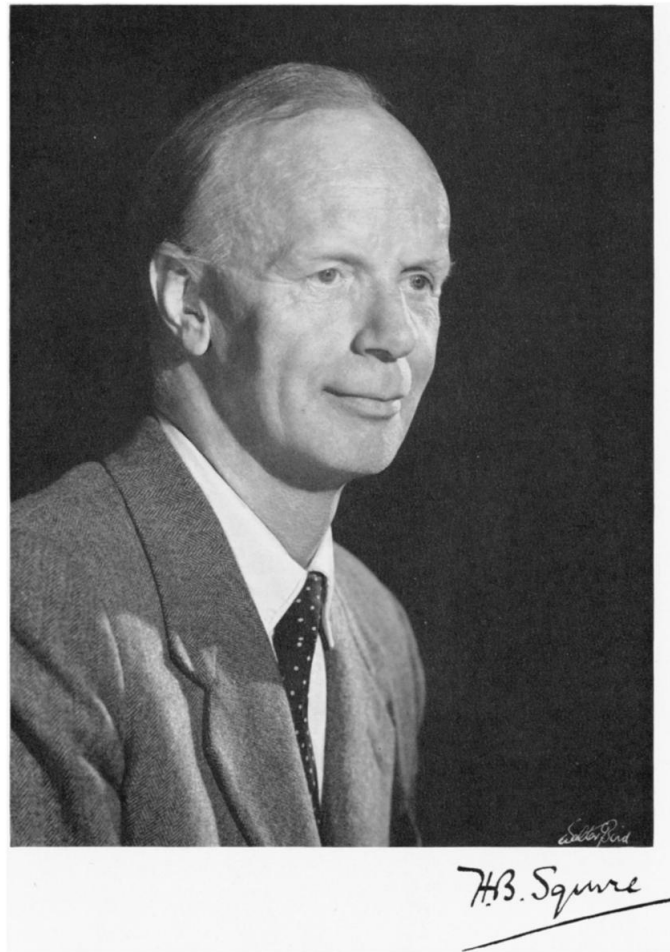


Figure 3: Herbert Brian Squire FRS (1909–61) was a specialist in fluid mechanics, and like Faber, he spent some time in Göttingen as a young man. His career was spent at the Royal Aircraft Establishment, the University of Manchester, the National Physical Laboratory, and Imperial College. [Figure needs to be cropped]



Figure 4: Nick Trefethen FRS is Professor of Numerical Analysis at the University of Oxford. His college, Balliol, is the same one where Squire was an undergraduate. [Figure needs to be cropped]